SPARSE CHANNEL ESTIMATION AND TRACKING FOR UNDERWATER ACOUSTIC COMMUNICATIONS

by

Danqing Yin

Submitted in partial fulfillment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
December 2016

Table of Contents

List of Tables

List of Figures

Abstract

This thesis focuses on estimation and tracking of underwater acoustic channels. A model of underwater acoustic channel suitable to study the channel estimation problem is constructed. The model is used together with a Bellhop acoustic wave propagation simulator to generate channels for various propagation environments, depths and distances, and we studied their sparseness and variation properties. Finally a number of estimation methods such as basis pursuit and approximate message passing are adapted to channel estimation problems and tested on a number of channels. Window-based and Kalman-filter based channel tracking is also studied. A detailed comparison of estimation and tracking methods is presented.

List of Abbreviations and Symbols Used

AMP: Approximate message passing

AR: Auto-regression

AR-1: Fist order auto-regression

AWGN: Additive white Gaussian noise

BER: Bit error rate

BP: Basis pursuit

BPDN: Basis pursuit de-noising

CAMP: Complex approximate message passing

CDF: Cumulative density function

CRLB: Cramer-Rao lower bound

CS: Compressed sensing

DFT: Discrete Fourier transform

FFT: Fast Fourier transform

Hz: Hertz

ICI: Inter-carrier interference

IDFT: Inverse discrete Fourier transform

i.i.d.: Independent and identically distributed

ISI: Inter-symbol interference

KF: Kalman filter

LS: Least squares

LTI: Linear time-invariant

MF: Matched filter

MP: Message passing

MSE: Mean square error

OFDM: Orthogonal frequency-division multiplexing

OMP: Orthogonal message passing

PDF: Probability density function

RC: Raised-cosine

RDR: Relative decay rate

RIP: Restricted isometry property

RLS: Recursive least squares

RRC: Root raised cosine

RT: Relatively thresholding

SNR: Signal-to-noise ratio

SOS: Sum of sinusoid

UUVs: Unmanned underwater vehicles

UWA: Underwater acoustic

B: Bandwidth

f: Carrier frequency

$L_{\max}$: Maximum delay in discrete time

T_{coh}: Coherence time

T_s: Sampling time

α: Shaper parameter of exponential form Doppler spectra

β: Roll-off factor of raised cosine filter

δ: Under-sampling parameter

ρ: Sparsity parameter

$\tau_{\max}$: Maximum delay in continuous time

$\tau_{\min}$: Relative minimum delay in continuous time

$\mathbf{A}$: Sensing matrix

$\mathbf{I}$: Identity matrix

$\mathbf{A}^h$: Hermitian transpose of matrix $\mathbf{A}$

$\mathbf{A}^T$: Transpose of matrix $\mathbf{A}$

A_{aj}^*: Conjugate of element A_{aj} in ath row and jth column of matrix $\mathbf{A}$

Acknowledgements

Thanks to my supervisor Dr. Dmitry Trukhachev for his patience and encouragement.

Chapter 1

Introduction

The area of underwater acoustic (UWA) communications has been growing rapidly over the recent years. Its origin can be traced back to World War I when the echo detection scheme was first explored, and its first application, the underwater telephone was developed in 1945 for communications with submerged submarines [1]. Apart from military use, nowadays, UWA acoustic communications finds applications in environmental monitoring, unmanned underwater vehicles (UUVs), oilfields exploration, etc.

Underwater channels are quite different from the classical wireless channels. Electromagnetic waves can only propagate for very short distances underwater because water is a conductive medium with very high absorption [2]. Hence low-frequency electro-magnetic communications underwater is only effective at distances of a few tens of meters. Similarly optical communications beyond few hundreds of meters is also impossible. Besides high attenuation optical communications underwater suffers from the scattering effect and requires high precision positioning alignment of the transmitter and receiver [3]. As a consequence, acoustic waves are generally the best choice for underwater wireless communications.

In general, however, UWA communications is much more challenging than radio communications. Major difficulties come from the properties of UWA channels, including multi-path propagation, long delay spread, and the Doppler effect. In addition, high frequency acoustic signals also experience high attenuation limiting the available communication bandwidths and as a consequence decreasing achievable data throughput. The multi-path propagation is mainly caused by the reflections of acoustic waves on the sea surface and bottom. The fact that the delay spread is long stems from the slow sound speed in the water (around 1500 m/s) compared to the speed of light. This make UWA channels frequency-selective and may result in significant inter-symbol interference (ISI) complicating the design of efficient transmitters and

receivers. The Doppler effect, on the other hand, leads to the time-varying behavior of UWA channels which is mainly due to the surface motion (Doppler spread) and the relative motion between transmitters and receivers (Doppler shift). Doppler spread makes UWA channels time-selective and therefore, the channel variations need be tracked over time. Hence, the channel is both time and frequency selective.

Many experiments [4] confirm that the Doppler spectrum of UWA channel may be closely approximated by a stretched exponential, which is different from the Jake's spectrum in radio channels. The channel coherence time is closely related to the inverse of the Doppler spread and for UWA channels, the product of the delay spread $\tau_{\max}$ and Doppler spread is often larger than for wireless channels. In case this product is close to unity or exceeds, then the channel is overspread. In an overspread UWA channel, the channel information is already changed when the last signal echo arrives with the last path. Therefore, channel estimation and tracking as well as coherent communications is impossible in such a scenario.

Typically an UWA channel's multi-path profile is sparse in nature [5]. That means there are just a few multi-path arrivals in a long delay span. This sparse structure suggests that estimation techniques geared to sparse signals may improve the performance of channel estimator and tracker. While traditional algorithms, like Least Squares (LS), take all the channel taps into account, then the estimation of many insignificant taps may add the burden of computation complexity and degrade the estimation accuracy. Hence, the motivation of this thesis is to learn more about actual sparsity in UWA channels and find out if the sparse structure can really give advantages to estimation and tracking methods.

The channel estimation methods used in this thesis are training-based, that is, some known pilot signals are sent first to obtain the channel information prior to transmission of actual data. A number of sparse estimation techniques available from the literature are being examined. These include Matching Pursuit (MP), a greedy algorithm to find a sparse solution over a set of dictionary [6], and its improved version Orthogonal Matching Pursuit (OMP), which also enforces orthogonality of residual errors at every step [7, 8]. In this thesis we also apply an alternative approach is called Approximate Message Passing (AMP), which is an iterative thresholding algorithm related to message passing on graphs [9] to channel estimation. In particular

we adapt the Complex Approximate Message Passing (CAMP) [10] version of the AMP to channel estimation and compare its performance to the matching pursuit algorithms both for time and frequency domain based estimation. Specifically we study a relationship between the sparsity level of the channel and the percentage of pilot bits relative to the data bits that needs to be used for reliable estimation. The details will be described in Chapter 3.

Since the UWA channel is time varying, it needs to be re-estimated frequently or, alternatively, its variation needs to be tracked over time. Based on this observation there are two approaches which address the time variability. The first is the window-based tracking method. It is based on selecting a window of channel observations such that the channel variation over the window is not too strong, then we estimate the channel using the observations to receive the communicated data, and then re-estimating it based on the new window again [5]. The other method first constructs an autoregressive (AR) model of the channel to characterize its time-varying behavior and then uses a channel tracker to track the channel. This technique has been explored in [11], and related research can be found in [12, 13, 14, 15]. With the latter approach we first send a few pilot signals to obtain the initial channel information and calculate the AR model coefficients, and then Kalman filter is applied to track the channel taps. The accuracy of the initial AR model can greatly influence the tracking performance. Therefore, the accuracy of the initial channel estimation is crucial. Both approaches are implemented in this thesis and compared.

There are three main contributions of this thesis.

First, a theoretic background for a basic dynamic channel model is outlined and then the model is derived, tested and used to generate UWA channels for estimation and tracking. The model incorporates the sum-of-sinusoid (SOS) technique presented in [16] to generate the dynamic variation of the analog channel taps and uses transmit-received filtering and sampling to generate discrete channel taps (in both time and delay domain). The model incorporates channel variation statistics based on Qarabaqi and Stojanovic's work [17]. The model introduced in this thesis possesses the important features required to design and testing of communications systems. At the same time it does not rely on a complicated set of parameters many of which are only known from experiments in specific underwater environments [17].

Second, an extensive study of UWA channel scenarios has been performed using Bellhop [18] software in combination with the derived channel model. It has been found that UWA channels are not always sparse, and in many cases various multi-path profile density levels are encountered. A specific notion of coherence time which fits channel estimation framework is introduced and feasibility of orthogonal frequency-division multiplexing (OFDM) signaling for various channel scenarios can be assessed.

Third, a number of channel estimation and tracking methods have been studied. For static channel estimation, we adapted AMP and CAMP algorithms for channel estimation, implemented OMP and compared their performances for particular cases and in terms of the phase transition curves. Different problem settings are studied. We have also explored the phase transition curves with the impact of filtering for OFDM systems which is a part of the work in paper [19]. For dynamic channel tracking, we compared the bit error rate (BER) performances of Kalman filter, recursive least squares (RLS), and OMP algorithms in OFDM setting with the channels generated by our model.

The outline of this thesis is as follows.

Chapter 2 presents the basic dynamic channel model with independent analog paths as its inputs and transmit-receiver matched filters are also included in this model. We used a single value α as a shape parameter of Doppler spetra to characterize the channel variation. In Section 2.3, different channel scenarios are studied and compared. The feasibility of OFDM communication systems is discussed for each scenario.

Next, Chapter 3 introduces two sparse estimation algorithms, OMP and AMP, and we compared them with respect to the phase transition performance. Different tuning methods and stopping criteria for those two algorithms are presented. In this chapter, we implemented time domain approach and frequency domain approach for sparse channel estimation. We then compared MSE performance of OMP, AMP, LS algorithms and some lower bounds in OFDM system setup. At last we discussed about the impact of filtering and resolution to sparse channel estimation.

Chapter 4 implemented two tracking mechanisms for dynamic channel estimation, one is window-based method for single carrier communication system and another is frequency domain method for OFDM communication system. Both of the tracking

mechanisms are sparse-aided, that is, we only track the dominant channel taps. The positions of the dominant channel taps are found in the initializing stage. Adaptive filter algorithms, RLS and Kalman filter are implemented for benchmark. MSE performance of RLS and BER performance of Kalman filter are presented.

Chapter 5 discussed about the analysis over the results in this thesis. We first present that in which scenario the channel is sparse or under-spread, and then give suggestions about which sparse algorithms is better for different channel scenarios. At last we make comparison between window-based tracking and frequency domain tracking and analyzed their pro and cons.

Finally Chapter 6 concludes this thesis and discusses the future work.

Chapter 2

Underwater Acoustic Channel Geometry and Characterization

2.1 A Mathematical Channel Model

A mathematical UWA channel model can provide us a useful tool for testing channel estimation and tracking algorithms. From the research in [17], UWA channels show random time-varying behavior in both large-scale and small-scale. However the larger-scale channel model mainly features the channel amplitude-delay profile given the channel geometry information and it varies slowly, usually including many wavelengths. Based on the large-scale channel model, the small-scale effects can be added to characterize the variations of instantaneous channel impulse responses. It is mainly involved in a few wavelengths of one path and caused by the signal scattering on the sea surface or the objects. One large-scale path can include numbers of intra-paths because of the scattering effect.

2.1.1 The Large-Scale Channel Model

In [17], the large-scale channel frequency response is given as

$$H(f) = H_0(f) \sum_{p=1}^{P} h_p e^{-j2\pi f \tau_p} \tag{2.1}$$

where f is the carrier frequency, $j^2 = -1$, P is the number of paths, h_p is the path amplitude and τ_p is the path delay. $H_0(f)$ is a low-pass filter assumed to underlie all the paths due to the large signal attenuations at high frequencies. This model determines the nominal paths between the receiver and transmitter as shown in Figure 2.1. In this example, there are 3 large-scale paths with two of them having reflection on the surface and the other one reflecting on the bottom.

Because of the long delay spread property and the signal attenuations along the transmission range, the UWA channel usually shows a sparse structure in the delay

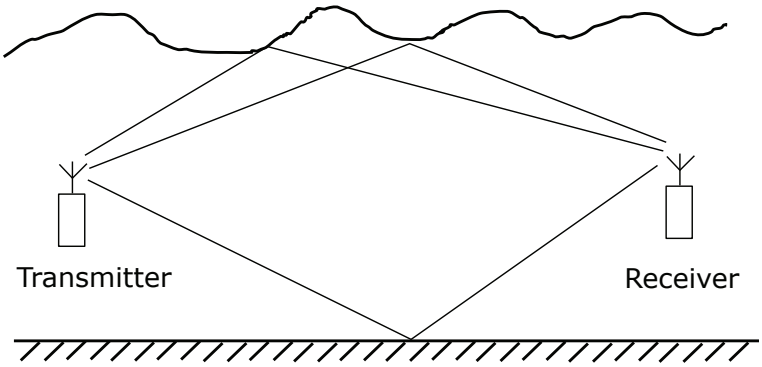


Figure 2.1: An example of large-scale effects of UWA channels.

domain in a long-range communication ($\geqslant$ 10 km [20]). If we apply the inverse Fourier transform to the Equation 2.1, and then we obtain a time domain expression as

$$h(t) = \sum_{p=1}^{P} h_p(t)\delta(t - \tau_p) \tag{2.2}$$

it is clearly to see that the channel impulse responses are composed of P echoes with P different amplitudes and arrival times. If we plot the amplitude of each arrival path versus its arrival time, then we obtain the channel amplitude-delay profile. Figure 2.2 is an example. We can see that there are 15 paths spanned in the delay domain, and the delay spread is around 0.013 seconds.

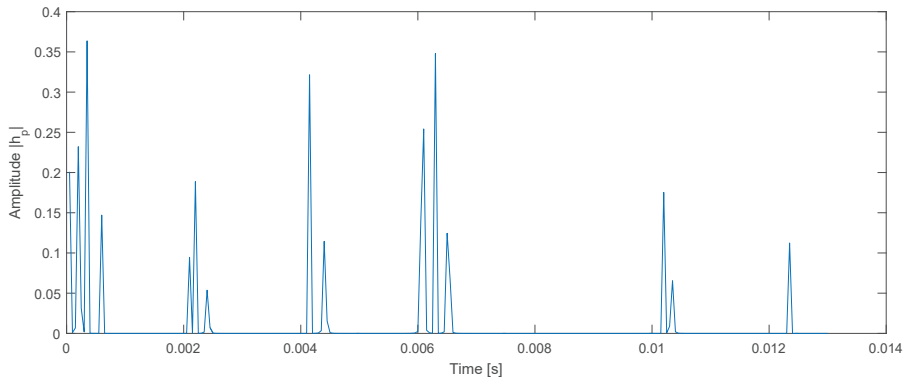


Figure 2.2: An example of amplitude-delay profile of a sparse UWA channel.

2.1.2 The Small-Scale Channel Model

For each large-scale path in Figure 2.1, the scattering effect can split it into many intra-paths, as shown in Figure 2.3. The scattering usually happens on the sea surface

because of its roughness. Now we call each path as a path bundle. Assume that, for path bundle p, there are S_p intra-paths, then adding this small-scale effect into Equation 2.1, we obtain

$$H(f) = H_0(f) \sum_{p=1}^{P} \sum_{i=1}^{S_p} h_{p,i} e^{-j2\pi f \tau_{p,i}} \tag{2.3}$$

where $h_{p,i}$ is the micro-path gain for path bundle p, $\tau_{p,i} = \tau_p + \delta\tau_{p,i}$ is the micro-path delay and $\delta\tau_{p,i}$ is the deviation of the micro-path delay assumed to be Gaussian distributed.

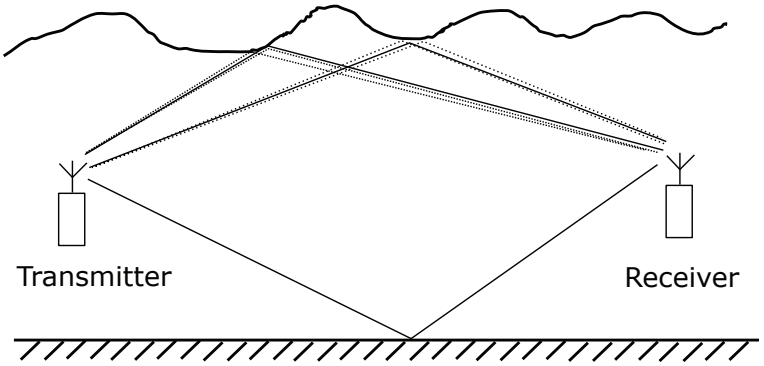


Figure 2.3: An example of small-scale effects of UWA channels.

Then the small-scale fading coefficient for path bundle p is defined as

$$\gamma_p(f) = \frac{1}{h_p} \sum_{i=1}^{S_p} h_{p,i} e^{-j2\pi f \delta\tau_{p,i}} \tag{2.4}$$

Substituting Equation 2.4 into 2.3, the overall channel frequency response is expressed as

$$H(f) = H_0(f) \sum_{p=1}^{P} h_p \gamma_p(f) e^{-j2\pi f \tau_p} \tag{2.5}$$

Since for each path bundle the intra-paths have similar distances, their amplitudes $h_{p,i}$ are also similar. However, the phase $2\pi f \delta\tau_{p,i}$ can change significantly resulting to the rapid variation of $\gamma_p(f)$. Hence, the statistical properties of $\gamma_p(f)$ determines the time-varying behavior of the overall UWA channel model. The details will be described in Section 2.2.

2.1.3 Basic Dynamic Channel Model

Observing the Equation 2.5, the low-pass filter $H_0(f)$ underlying all the paths makes paths dependent and therefore, the channel model is not wide-sense stationary. Hence, to generate the paths separately and more simply, we assume that the channel model consists of some independent impulses in conjunction with transmit and receiver filters, as shown in Figure 2.4, and we denote the whole blocks as basic dynamic channel model.

Figure 2.4: Block diagram of basic dynamic channel model.

The transmit-receiver filter must satisfy the Nyquist criteria [21] and it replaces the low-pass filtering effect of $H_0(f)$. Here we consider about the Root-Raised Cosine (RRC) filter pairs. Those independent impulses can be generated by the sum-of-sinusoids (SOS) model originally proposed in [22]. In the discrete time domain, each path $h_p[i] = h_p(iT_s)$ can be created according to

$$h_p[i]_{\text{Rayleigh}} = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} e^{j(\phi_m + 2\pi f_m i T_s)} \tag{2.6}$$

where each path are composed of M sinusoid waves and follows Rayleigh distribution. Here, T_s is the sampling time consistent with the bandwidth of transmit-receiver filter, ϕ_m are random phases uniformly distributed in the interval $[0, 2\pi]$ with initial phase ϕ_0 and f_m are the M Doppler frequencies of each path.

We can notice that in the SOS model, each path h_p is time-varying with the sample index i. To capture the time-varying behavior, we need the statistical distribution of Doppler frequencies f_m. From the experiments in [17] and [4], the Doppler spectrum of UWA channel is measured to be a stretched exponential shape. In formula, the Doppler spectrum can be expressed as

$$S(v) = \frac{1}{2\alpha} e^{-\frac{|v|}{\alpha}} \tag{2.7}$$

where v denotes the Doppler frequency, and α is the shape parameter that matches the Doppler spectrum to the exponential function. Hence, f_m can be drawn from

the distribution with probability density function (PDF) expressed as Equation 2.7. Derived from the inverse transform sampling lemma, the cumulative distribution function (CDF) of f_m is uniformly distributed in $[0, 1]$, which is expressed as

$$\begin{aligned} F(f_m) &= \int_{-\infty}^{f_m} S(v)dv \\ &= \int_{-\infty}^{f_m} \frac{1}{2\alpha} e^{-\frac{|v|}{\alpha}} \\ &= \frac{\mathrm{sgn}(f_m)}{2}(1 - e^{-\frac{|f_m|}{\alpha}}) \end{aligned} \tag{2.8}$$

Then let $F(f_m) = u$ be uniformly distributed in $[0, 1]$, we obtain

$$f_m = -\alpha \log(\mathrm{mod}(2u, 1))\mathrm{sgn}(1 - 2u) \tag{2.9}$$

therefore, f_m can be generated using Formula 2.9 with PDF expressed as Equation 2.7. Here α determines the exponential shape of Doppler spectrum $S(v)$, and larger Doppler spread is obtained with larger value of α.

To consider about the transmit-receiver filtering effect in Figure 2.4, assume that we transmit pulse trains $x(t) = \sum_{i=-\infty}^{\infty} x[i]q(t - iT_s)$ with sampling interval T_s, where $q(t) = g(t) * g(-t)$ is the convolution between transmit filter $g(t)$ and receiver filter $g(-t)$. Here, $q(t)$ is the convolution between two RRC filters which leads to a root-cosine (RC) filter. The formula is given as

$$q(t) = \frac{\cos(\pi\beta\frac{t}{T_s})}{1 - (2\beta\frac{t}{T_s})^2} \times \frac{\sin(\pi\frac{t}{T_s})}{\pi\frac{t}{T_s}} \tag{2.10}$$

where β is the roll-off factor determining the bandwidth B of $Q(f)$ which is the Fourier transform of $q(t)$, and $B = \frac{1+\beta}{T_s}$. $q(t)$ has zero-crossing at $t = kT_s, k = 0, \pm1, \pm2, \ldots$. If there is only one path of the channel $h_1(t)$ with delay τ_1, then the received signal is

$$y(t) = \sum_{i=-\infty}^{\infty} A_1 h_1(t) x[i] q(t - iT_s - \tau_1) + n(t) \tag{2.11}$$

where A_1 is the averaged path amplitude and $n(t)$ is the additive white Gaussian (AWGN) noise. Sampling at time $t = jT_s, j = 0, 1, 2, \ldots$, we obtain the sampled received signals as

$$y[j] = y(jT_s) = \sum_{i=-\infty}^{\infty} A_1 h_1(jT_s) x[i] q(jT_s - iT_s - \tau_1) + n(jT_s) \tag{2.12}$$

Denote the sample of $h(t)$ at time $t = jT_s$ as $h[j]$, and denote $n(jT_s)$ as $n[j]$, we have

$$y[j] = \sum_{i=-\infty}^{\infty} A_1 h_1[j] x[i] q\left((j-i)T_s - \tau_1\right) + n[j] \tag{2.13}$$

Combining the channel tap $h_1[j]$ with filtering function $q(t)$, we obtain the equivalent formula

$$\begin{aligned} y[j] &= \sum_{i=-\infty}^{\infty} f[j,l] x[i] + n[j] \\ &= \sum_{l=1}^{L} f_j[l] x[j-l] + n[j] \quad \text{(assume } L \text{ taps in a causal system)} \end{aligned} \tag{2.14}$$

where $j - i = l$, and

$$f_j[l] = A_1 h_1[j] q(lT_s - \tau_1) \tag{2.15}$$

If we have multiple paths, we generalize the formula 2.15 as

$$f_j[l] = \sum_{p=1}^{P} A_p h_p[j] q\left(lT_s - \tau_p\right) \tag{2.16}$$

Equation 2.14 is the tapped-delay line model, and then we have a equivalent block diagram of Figure 2.4 as shown below.

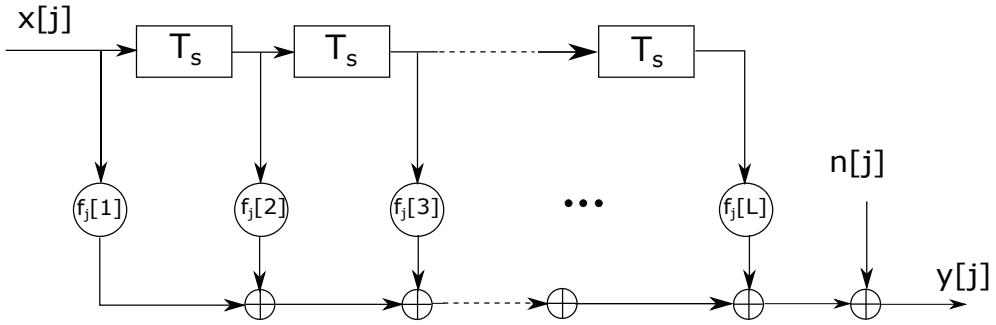


Figure 2.5: A tapped-delay line model.

In summary, the process of generating the basic dynamic channel model with filtering effects is as following.

1. Generate each path h_p based on Equation 2.6 with Doppler frequency f_m created by Equation 2.9. The averaged path amplitudes A_p is generated based on the channel geometry information which is assumed to known before applying this model. The details about how to generate A_p will be described in Section 2.3.

2. Add the filtering effect to each path following Equations 2.16 and 2.10.

There is still a unknown parameter α need to be set in Equation 2.9, which is a shape parameter of Doppler spectrum. It determines the Doppler spread describing how fast the channel will vary. The next section will talk about how to set the value α to make it represent the small-scale channel variation correctly.

Here is an example of generating the basic dynamic channel model. The inputs are

1) SysPara.B = 20000 (System bandwidth in Hz)

2) SysPara.fc = 30000 (Carrier frequency in Hz)

3) SysPara.fostime = 1 (Oversampling factor in time domain)

4) SysPara.fostau = 1 (Over sampling in delay domain)

5) SysPara.beta = 1 (Rolloff factor of transmitter-receiver filter)

6) SysPara.taua = [0 0.8 1.5 2.75 10.25 10.75 11.8 20.5 21.8 30.15 31.25 32.35 50.75 51.45 61.5]/5$\times 10^{-3}$ (Analog path delays in seconds)

7) SysPara.Pow = [0.8 0.7 0.9 0.3 0.5 0.7 0.1 0.8 0.2 0.6 0.8 0.4 0.5 0.3 0.2] (Analog path powers)

8) SysPara.M = 500 (The number of sinusoid waves in SOS model)

9) SysPara.a = 1 (Shape parameter of the stretched exponential Doppler spectra)

10) SysPara.del_f = 0(Dopper shift)

11) SimPara.NumSamp = 2000 (Total number of samples to generate in discrete time domain)

12) SimPara.Lmax = 260 (Maximum delay sampled in discrete time domain)

The output is a channel tap matrix $\mathbf{H}$. Each entry $\mathbf{H}_{ij}$ denotes the amplitude of the channel tap in ith sampling time and jth delay, for $i = 1, 2, \ldots,$ SysPara.NumSamp and $j = 1, 2, \ldots,$ SysPara.Lmax. An example of its plot shows in Figure 2.6.

There are 15 analog paths in this example, and each path has the same Doppler spread since the shape parameter SysPara.a is set to be the same for each path. Taking two channel taps with largest amplitude as examples, we plot the histogram of their Doppler frequencies and obtain the approximate Doppler spectra in Figure 2.7.

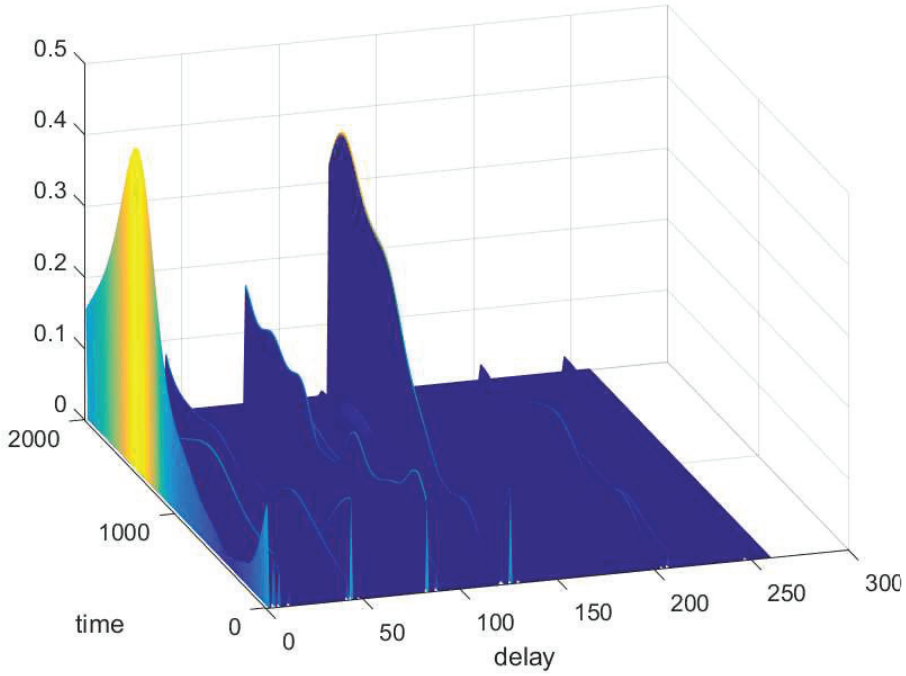
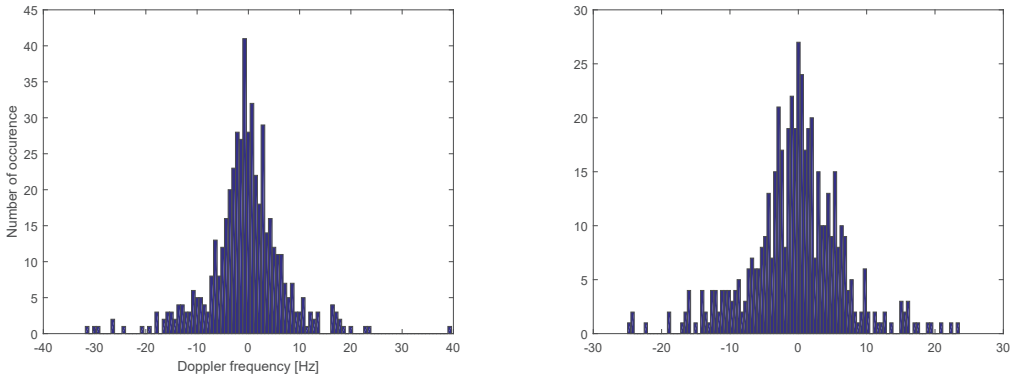


Figure 2.6: An 3-dimensional plot example of a realization of basic dynamic channel model. The number of analog paths is 15 and the shape parameter α is the same for every path.



(a) Doppler spectra of the channel tap with largest amplitude.

(b) Doppler spectra of the channel tap with second largest amplitude.

Figure 2.7: An example of approximate Doppler spectra of channel taps generated by basic dynamic channel model.

2.2 Time Variation of the Channel

In Section 2.1.2, the small-scale effects of UWA channels are characterized by the fading coefficient $\gamma_p(f)$. Because of the change of its random phase $2\pi f \delta\tau_{p,i}$, $\gamma_p(f)$ is also a function of time, hence, for convenience we rewrite the Equation 2.5 as

$$H(f,t) = H_0(f) \sum_{p=1}^{P} h_p \gamma_p(f,t) e^{-j2\pi f \tau_p} \tag{2.17}$$

where $\gamma_p(f,t)$ is the main contributor of channel variations for each path. In [17], its time autocorrelation function is expressed as

$$\begin{aligned} R_p(f,\Delta t) &= \mathbb{E}[\gamma_p(f,t)\gamma_p^*(f,t+\Delta t)] \\ &\approx 2\sigma_p^2(f) e^{-\pi B_p(f)|\Delta t|} \end{aligned} \tag{2.18}$$

where $2\sigma_p^2(f)$ and $B_p(f)$ are the variance and the effective Doppler bandwidth (Doppler spread) of $\gamma_p(f,t)$ and they are large-scale parameters. The approximation is based on the assumption that the intra-path delays $\delta\tau_{p,i}(t)$ are Gaussian distributed, and obey a first-order autoregressive model. The authors in [17] have also defined

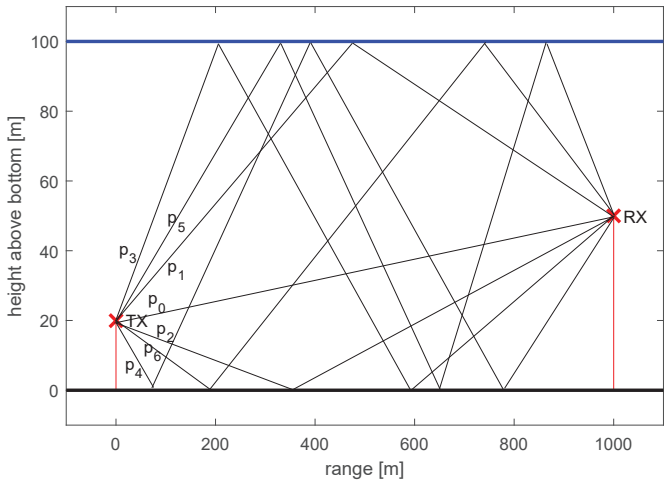
$$B_p(f) = (2\pi f \sigma_{\delta p})^2 B_{\delta p} \tag{2.19}$$

$$\sigma_p^2(f) \approx \frac{1}{2}\mu_p^2 S_p \left(1 - e^{-(2\pi f)^2 \sigma_{\delta p}^2}\right) \tag{2.20}$$

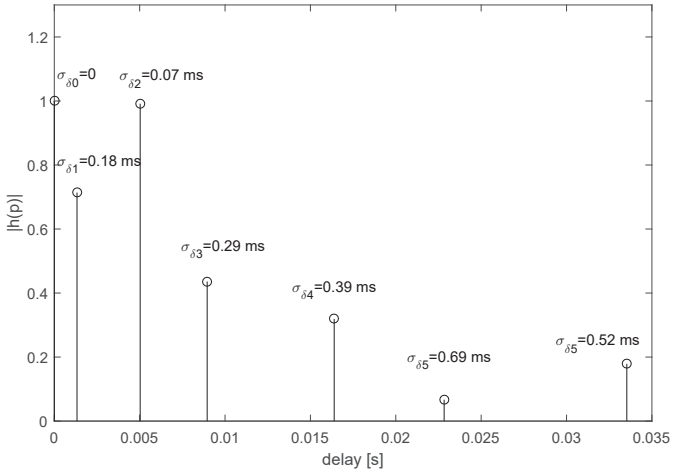
where $\sigma_{\delta p}^2$ and $B_{\delta p}$ are the variance and the 3-dB width of the power spectrum density of the intra-path delays of path p, μ_p is the mean value of the intra-path amplitudes, and S_p is the number of intra-paths. We assume that S_p intra-paths form one large-scale path. Based on Equations (2.18) - (2.20), one can see that the time autocorrelation function $R_p(f,\Delta t)$ is mainly determined by the small-scale parameters $\sigma_{\delta p}$ and $B_{\delta p}$, and hence by the variations in propagation delays of these many intra-paths. These, in turn, are heavily affected by the sea stat. It is assumed that $B_{\delta p}, \mu_p$, and S_p are all constant for all large-scale paths. The variance parameter $\sigma_{\delta p}^2$ is determined by the geometry of the channel, and in turn, depends on the number of reflections n_{sp} on the surface and at the bottom, n_{bp}, and the angle of arrival θ_p. If we denote the intra-path delay variances on the surface and bottom by σ_s^2 and σ_b^2 respectively, we have

$$\sigma_{\delta p}^2 = \frac{1}{c}(2\sin\theta_p)^2 (n_{sp}\sigma_s^2 + n_{bp}\sigma_b^2) \tag{2.21}$$

Different large-scale paths experience different numbers of bounces on the surface and bottom that result in different values of $\sigma_{\delta p}^2$. Therefore $\sigma_{\delta p}^2$ is path-dependent. An example of a shallow water propagation scenario is illustrated in Figure 2.8.



(a)



(b)

Figure 2.8: An example of shallow-water channel geometry and its amplitude-delay profile. Figure (a) is an example of shallow-water wave propagation and (b) is its corresponding amplitude-delay profile with different value of $\sigma_{\delta p}$ on each path.

We can see that there are 7 paths in this example. The path p_0 is a direct path, and all other paths either have reflections on the surface, the bottom or both. Figure 2.8(b) shows the channel amplitude-delay profile without filtering. Different

numbers of bounces and different arrival angles result in different variances of intra-path delays $\sigma_{\delta p}^2$. In general, the higher the number of bounces of a path, the larger the variance of the intra-path delays and attenuation of the path.

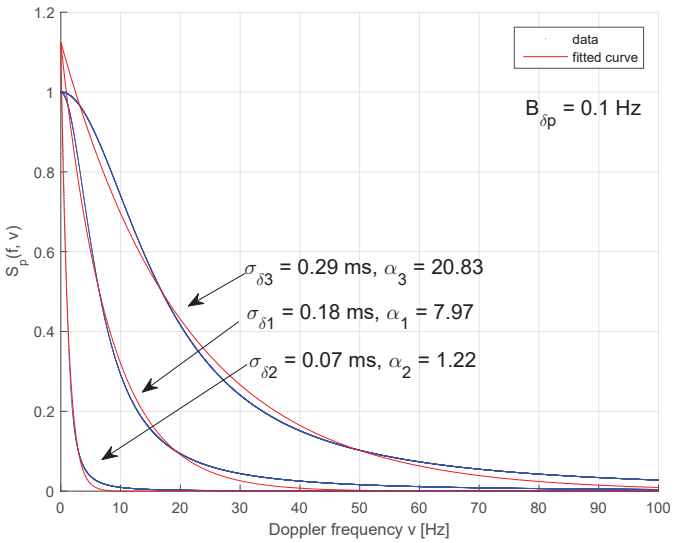
To understand the relationship between the Doppler spectrum and the small-scale parameters, we calculate the Doppler spectrum of path p as the Fourier transform of the autocorrelation function Equation (2.18)

$$\begin{aligned} S_p(f,v) &= \int_{-\infty}^{\infty} R_p(f,\Delta t) e^{-j2\pi v\Delta t} \mathrm{d}\Delta t \\ &= \int_{-\infty}^{0} 2\sigma_p^2(f) e^{\pi B_p(f)\Delta t} e^{-j2\pi v\Delta t} \mathrm{d}\Delta t + \int_{0}^{\infty} 2\sigma_p^2(f) e^{-\pi B_p(f)\Delta t} e^{-j2\pi v\Delta t} \mathrm{d}\Delta t \\ &= \frac{4\pi\sigma_p^2(f) B_p(f)}{\pi^2 B_p^2(f) + (2\pi v)^2} \end{aligned} \tag{2.22}$$

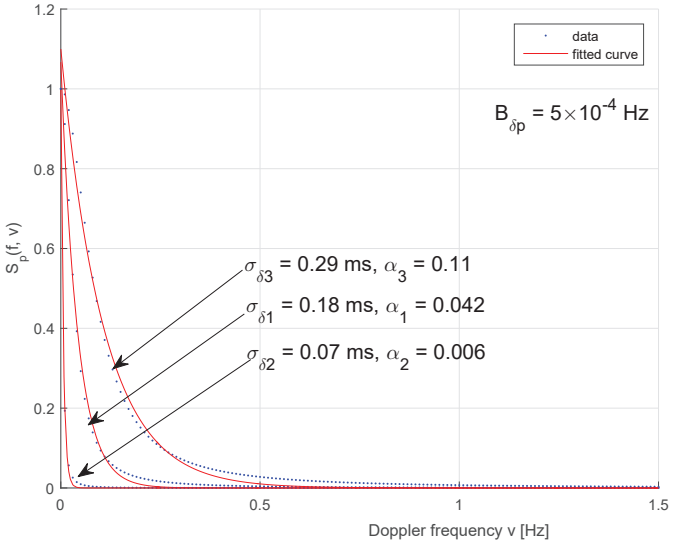
where v is the Doppler frequency and f is the carrier frequency. In the small-scale channel model the Doppler spectrum is a function of the carrier frequency f, and different paths have different Doppler spectrum due to Equations (2.19) and (2.20). To fit Equation (2.22) to an exponential form as Equation (2.7), we use least squares curve fitting method. For the example in Figure 2.8, we choose paths $p = 1, 2, 3$ to demonstrate our results. We set the carrier frequency to $f = 10$ kHz, $\mu_p = 0.0025$ and $S_p = 20$ (we use the parameter setting in [17]). The Doppler spectrum expressed by Equation (2.22) and its fitted exponential curve are shown in Figure 2.9 for positive frequencies.

For paths $p = 1, 2, 3$, the parameter α of the fitted curve is computed and shown in Figure 2.9. On the one hand, we can see that different paths with different values of $\sigma_{\delta p}$ are characterized by different values of α. The total Doppler spectrum is the summation of the spectra of all the larg-scale paths. The total Doppler spread is typically dominated by the path which has the largest value of $\sigma_{\delta p}$. If we want to use the stretched exponential function (2.7) with a single constant parameter α for all paths, it is reasonable to choose the value of α based on the worst significant path which has the largest Doppler spread.

Comparing Figures 2.9(a) and (b), we note that $B_{\delta p}$ has a significant impact on the Doppler spectrum as well. We choose two candidate values of $B_{\delta p}$ based on the experiments in [17]. It can be seen that larger $B_{\delta p}$ gives us a broader Doppler spectrum, which indicates that large Doppler spread of intra-paths can lead to larger



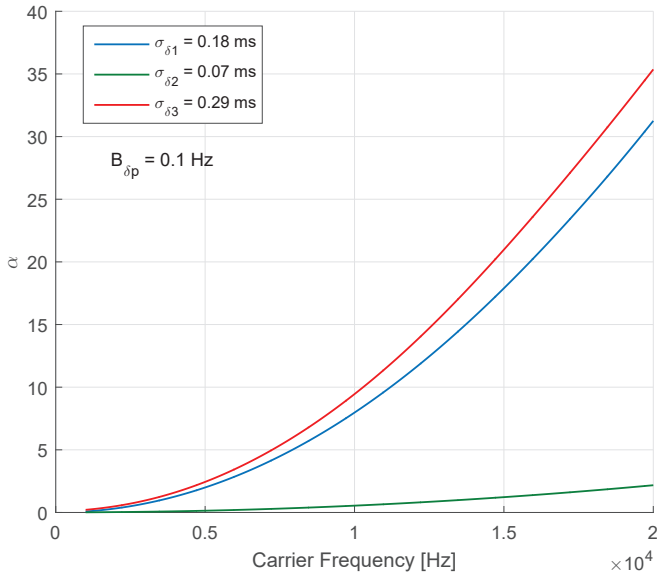
(a) Doppler spectrum for $B_{\delta p} = 0.1$ Hz



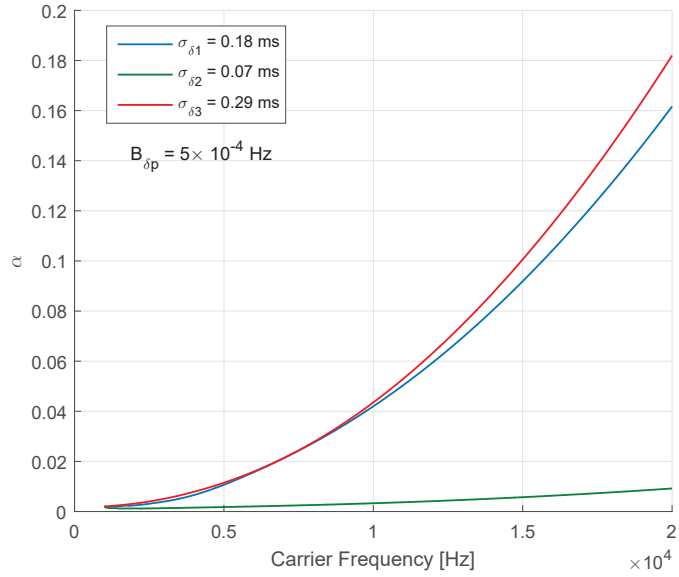
(b) Doppler spectrum for $B_{\delta p} = 5 \times 10^{-4}$ Hz

Figure 2.9: Doppler spectrum of paths p_1, p_2 and p_3 in Figure 2.8 with carrier frequency $f = 10$ kHz, the mean of intra-paths amplitudes $\mu_p = 0.0025$ and the number of intra-paths $S_p = 20$. The red curves are their fitted exponential curves.

Doppler spread of large-scale paths. Using this approach and applying different carrier frequencies, we find out how the value of α increases with carrier frequency (see Figure 2.10). Broadening of the Doppler spectrum at high frequencies is observed.



(a) $B_{\delta p} = 0.1$ Hz



(b) $B_{\delta p} = 5 \times 10^{-4}$ Hz

Figure 2.10: The relationship between α and the carrier frequency f for the channel in Figure 2.8. The channel parameters are the same as in Figure 2.9.

In summary, the small-scale parameter $\sigma_{\delta p}$ and $B_{\delta p}$ determine the large-scale parameters σ_p and B_p, and in turn, affects the Doppler spectrum of each large-scale path. For choosing one single value of α to represent the overall Doppler spread of the channel to put into the SOS model (Equation 2.9), the reasonable solution is

to choose the largest α corresponding to the broadest Doppler spread among all the large-scale paths.

2.3 Channel Geometry

In this section, we consider different underwater acoustic channel topologies, and apply the Bellhop software for acoustic wave propagation [18] to compute the multi-path profiles for these channels. Incorporating the effect of transmit-receiver filtering, we discuss the resulting channel parameters important for communications and study the effects on the channel estimation.

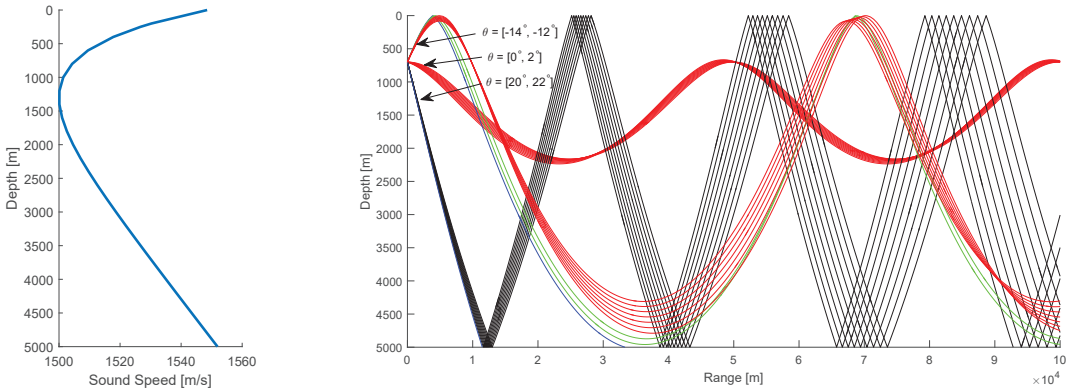
The inputs of the Bellhop simulator are the transmitter-receiver distance, transmitter-receiver depth, water depth, the sound speed profile and the signal frequency. The outputs of the simulator provide the delay-amplitude profile of the channel. We use the Bellhop simulator to generate the nominal channel taps, which is the large-scale paths. We then add transmit-receiver filtering and the Doppler effect to characterize the time variation of the channel when we generate the SOS channel model in Figure 2.4.

We simulate five UWA channel scenarios which are often used for UWA communiations, including long-range deep-water channels, long-range shallow-water channels, short-range shallow-water channels, medium-range deep-water channels and arctic channels.

2.3.1 Long-Range Deep-Water Channels

We use the Bellhop software to evaluate the wave propagation a deep-water channel at depth $d = 5000$ m, with transmitter-receiver distance $l = 10$ km, and the depth of the transmitter and the receiver are up to 1000 m. In this scenario, we transmit with a carrier frequency $f = 2$ kHz, and bandwidth $B = 300$ Hz. The sound speed profile is set to be the Munk profile (see Figure 2.11(a)), and the angle θ of beams in Bellhop is from $-50°$ to $50°$. An example of ray tracing is shown in Figure 2.11(b). If we change the angle of the beams, the rays propagate differently. The red curves represent the rays that have no bounce at the bottom or surface, the black curves represent the rays that hit both boundaries, the blue ones only hit the bottom and the green ones only hit the surface. Therefore, if we put the receiver at different depths and ranges,

we will obtain different channel delay profiles, and the receiver may even lie in the shadow zone where no ray passes. Transmitting with different launch angles have impact on the distance of propagation paths. The path with higher launch angles will have more reflections and longer propagation distance. This will result in higher transmission loss. The reasons are the scattering effect on the sea surface and the absorption of energy by the water. For example, the paths with launching angle from 0° to 2° have the shortest propagation distance and do not have any reflections on the sea surface.



(a) Munk sound speed profile (b) Ray tracing as a function of depth and range (angle from −5° to 5°)

Figure 2.11: An example of sound speed profile in a deep-water channel and its corresponding ray tracing plot.

When we put the transmitter of depth $d_{\text{tx}} = 700$ m, and put 50 receivers uniformly up to depth 1000 m and distance 10 km, the number of delays we observe at different receiver depths d_{rx} is shown in Figure 2.12. The maximum delay $\tau_{\max}$ increases with d_{rx}, which has a impact on the difficulty of communication. There is no channel tap when the receiver is above 700 m due to the shadow zone.

To understand the time variation behavior of this long-range deep-water channel after filtering we applied different values of α to obtain different levels of Doppler spread (see Equation (2.7)), and then calculate the corresponding coherence time T_{coh} numerically, which is based on percentage of change of the channel path amplitude $\mathbb{E}_t\left(\frac{||h_p(t)-h_p(t+T_{\text{coh}})||}{||h_p(t)||}\right) \leq \theta$. We then obtain the maximum T_{coh} corresponding to a threshold of $\theta = 10\%$.

Here, we applied the same value of α for all the taps. The coherence time T_{coh} is

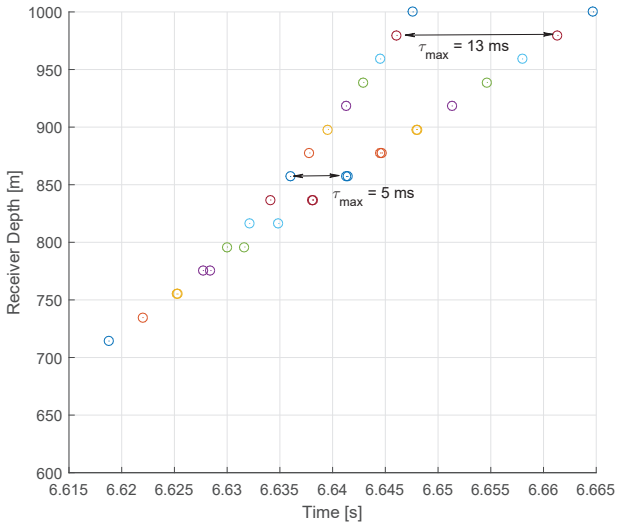


Figure 2.12: Different channel delay profiles for different receiver depths in a long-range deep-water channel.

a useful parameter when implementing a transmission system. Assume the sampling interval is T_s, and the number of discrete-time channel taps is $L_{\max}$, and the number of carries in an OFDM system is N, then T_{coh} must satisfy

$$(L_{\max} + N)T_s < T_{\text{coh}} \tag{2.23}$$

in order to avoid inter-channel interference (ICI). $L_{\max}$ is also the minimum number of samples of the OFDM cyclic prefix. Therefore, with different values of α, we can calculate the coherence time T_{coh} and see whether the channel is overspread or not w.r.t an OFDM frame, and in turn decide whether the OFDM system can be implemented on a specific channel without complex ICI compensation. For example, if a channel is overspread, which means the maximum delay $\tau_{\max} > T_{\text{coh}}$, then N will be negative. We conclude our results in Table 2.1 choosing the delay profile where $\tau_{\max} = 17$ ms as example.

We can see that the channel is underspread when $\alpha = 0.003, 0.03$ and OFDM can be implemented. An important note is that we choose the value of α from Figure 2.10 with $f = 2$ kHz. α_1 is taken from Figure 2.10(b) which is an example of a calm channel, and α_2 and α_3 are taken from Figure 2.10(a) corresponding to a fast varying channel. T_{coh} can vary slightly with different paths and we choose the smallest value among all the paths.

Long range deep water (l = 10 km)			
$f = 2$ kHz, $B = 300$ Hz, $\tau_{\max} = 17$ ms, $L_{\max} = 9$			
	Coherence time T_{coh}	Spreading factor $\frac{\tau_{\max}}{T_{\text{coh}}}$	Number of OFDM sub-carriers N
$\alpha_1 = 0.003$	1.5 s	0.0113	441
$\alpha_2 = 0.03$	0.22 s	0.0772	57
$\alpha_3 = 0.5$	0.016 s	1.06	<1

Table 2.1: Summary of time variation of long-range deep-water channel (transmission range $l = 10$ km).

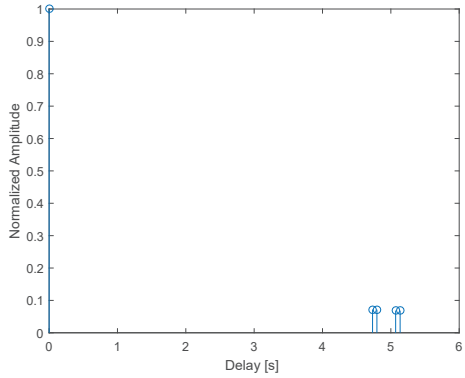
We then apply the same parameters but change the transmitter-receiver distance to $l = 50$ km for the same simulation. We put 50 receivers uniformly at different depths and obtained 50 different channel multi-path profiles. Figure 2.13 demonstrates the resulting multi-path profiles corresponding to three different depths of receivers. The sub-figures on the left hand side correspond to the actual channel impulse responses generated by the Bellhop with path delays measured in seconds. The sub-figures on the right are the channel tap-delay line coefficients obtained after the application of the transmit-receive filtering assuming that no Doppler effect is added. Root-raised cosine filters with $\beta = 1$, $B = 300$ Hz, are applied at both transmitter and receiver and sampling with sampling interval $T_{\text{s}} = 1/B$ is done at the receiver. Because of the sidelobes introduced by the filtering, we set a cut-off at 10^{-3} for the path amplitudes of the paths shown in the right hand side sub-figures.

The common property of the figures on the left is that the multi-path profiles are all sparse with only few delays. However, they potentially show different difficulty levels of channel estimation, and here we choose three candidates to demonstrate our observations.

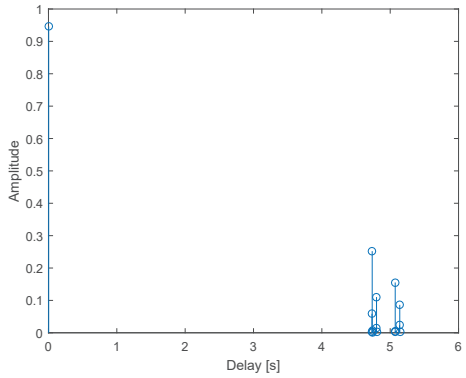
Let us denote the minimum delay as $\tau_{\min}$, the time difference between two closest paths, and the operation bandwidth of the transmitter-receiver filter as B. Based on the super-resolution theory [19], the necessary and sufficient condition to resolve all the channel taps in the discrete-time domain is

$$B > \frac{4}{\tau_{\min}} \tag{2.24}$$

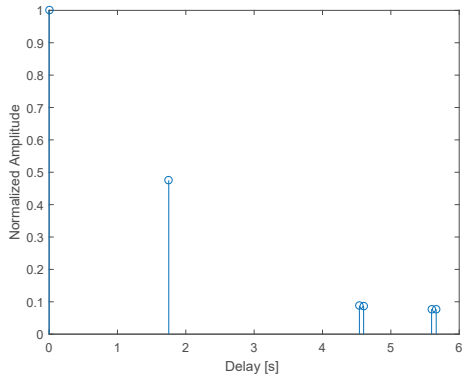
and we denote the minimum operation bandwidth as $B_{\min} = \frac{4}{\tau_{\min}}$.



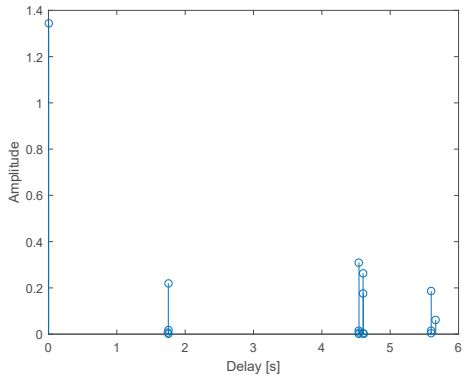
(a) Channel delay profile with receiver depth $d_{\text{rx}} = 510$ m, $B_{\text{min}} = 63$ Hz



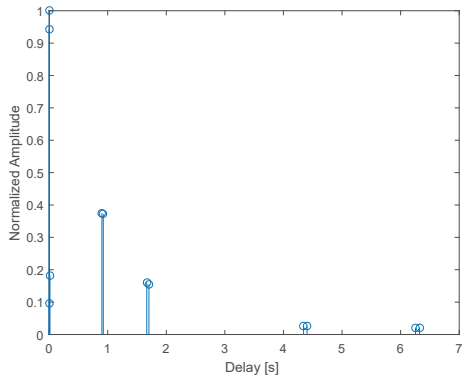
(b) Channel delay profile after filtering effect with receiver depth $d_{\text{rx}} = 510$ m



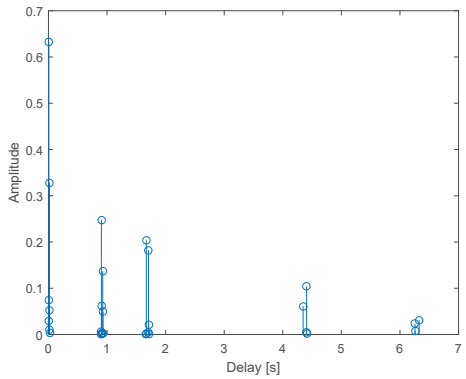
(c) Channel delay profile with receiver depth $d_{\text{rx}} = 1530$ m, $B_{\text{min}} = 65$ Hz



(d) Channel delay profile after filtering effect with receiver depth $d_{\text{rx}} = 1530$ m



(e) Channel delay profile with receiver depth $d_{\text{rx}} = 2755$ m, $B_{\text{min}} = 2766$ Hz



(f) Channel delay profile after filtering effect with receiver depth $d_{\text{rx}} = 2755$ m

Figure 2.13: Different channel delay profiles in a deep-water long-range scenario (transmission range $l = 50$ km).

Comparing these three sub-figures on the left, we can see that the delay spread increases with receiver depth. In Figure 2.13(a) and (c), we can clearly see all the taps,

but in Figure 2.13(e) some of the taps are very close to each other, which means that the minimum delay $\tau_{\min}$ is very small, and therefore, a larger bandwidth is required to satisfy the condition (2.24). Here, we indicate the minimum bandwidth $B_{\min}$ in the caption of the sub-figures, and we can see from Figure 2.13(b)(d)(f), $B = 300$ Hz is sufficient to resolve all the channel taps.

The time variation behavior is concluded in Table 2.2. Because all sub-figures of Figure 2.13 on the left hand side have similar values of $\tau_{\max}$, we take Figures 2.13(a)(b) as an example.

Long range deep water (l = 50 km)			
$f = 2$ kHz, $B = 300$ Hz, $\tau_{\max} = 5.14$ s, $L_{\max} = 1541$			
	Coherence time T_{coh}	Spreading factor $\frac{\tau_{\max}}{T_{\text{coh}}}$	Number of OFDM subcarriers N
$\alpha_1 = 0.003$	1.95 s	2.64	<1
$\alpha_2 = 0.03$	0.213 s	24.13	<1
$\alpha_3 = 0.5$	0.013 s	395	<1

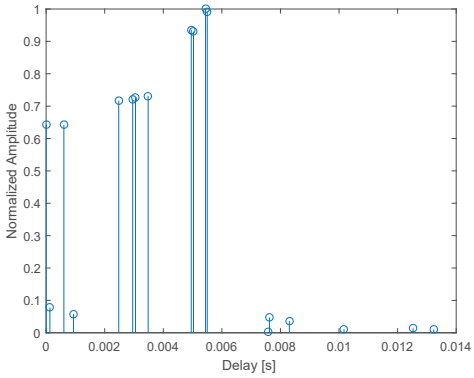
Table 2.2: Summary of time variation of long-range deep-water channels in Figure 2.13 (a)(b) (transmission range $l = 50$ km).

From Table 2.2, it can be seen that the spreading factors are larger than 1 for all the cases, therefore the channel is overspread and OFDM is not applicable without causing ICI.

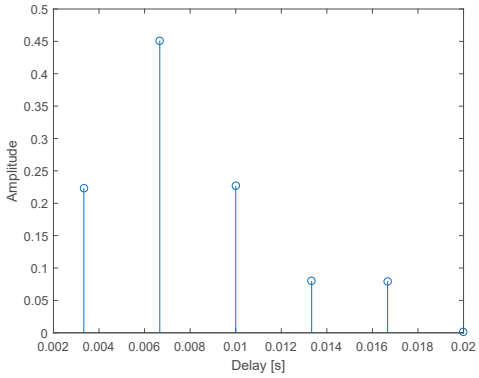
2.3.2 Long-Range Shallow-Water Channels

In a long-range shallow-water scenario, we set the channel depth $d = 100$ m, channel distance $l = 50$ km, the depth of the transmitter and receiver $d_{\text{tx}} = 50$ m, $d_{\text{rx}} = 12$ m. We transmit signals with carrier frequency $f = 2$ kHz and bandwidth $B = 300$ Hz. In the shallow water simulation, the sound speed profile is not needed, instead, the sound speed c in the water and the sound speed c_2 in the bottom are needed. Typically c is around 1500 m/s and it changes little based on temperature, pressure and salinity. The bottom sound speed c_2 is influenced by the hardness of the bottom, and c_2 increases dramatically with a harder bottom.

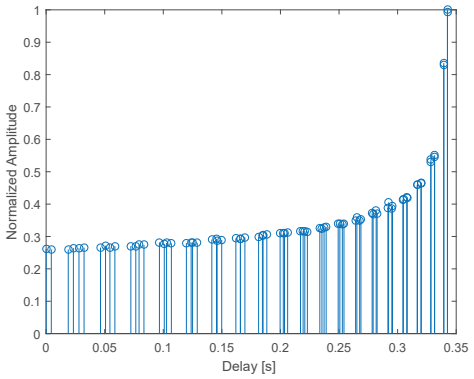
In Figure 2.14, we generated two cases of channel impulse responses, Figure 2.14(a) for the soft bottom case and Figure 2.14(c) for the hard bottom case. In both cases,



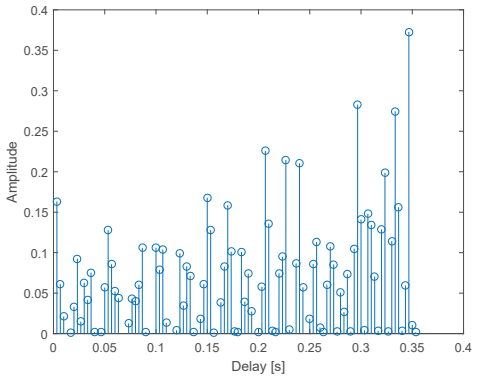
(a) Channel delay profile with soft bottom $c_2 = 1550$ m/s, $B_{\min} = 95$ kHz



(b) Channel delay profile after filtering effect with soft bottom $c_2 = 1550$ m/s



(c) Channel delay profile with hard bottom bottom $c_2 = 1650$ m/s, $B_{\min} = 200$ kHz



(d) Channel delay profile after filtering effect with hard bottom bottom $c_2 = 1650$ m/s

Figure 2.14: Different channel delay profiles in a shallow-water long-range scenario.

we set $c = 1548$ m/s. We can see that in shallow water with hard bottom, there are many more channel taps compared to the soft bottom case due to more reflections at a harder bottom. Comparing Figure 2.14 with the deep-water scenario in Figure 2.13, the channel impulse responses in deep water have higher sparsity level than those in shallow water and also have longer delays and larger minimum delay. We can see from Figure 2.14(b) and (d) that after filtering (the same filter parameters are applied as in Section 2.3.1), most of the channel taps cannot be resolved by a system bandwidth $B = 300$ Hz because of the small minimum delay.

Tables 2.3 and 2.4 summarize the time variation behavior of the long-range shallow-water channel. Applying the same approach as in previous sections, we can see that the channel is underspread for α_1 and α_2 in the soft bottom case, and OFDM can be implemented. However, in the hard bottom case, the channel is overspread.

Long range shallow water (soft bottom)			
$f = 2$ kHz, $B = 300$ Hz, $\tau_{\max} = 13.2$ ms, $L_{\max} = 6$			
	Coherence time T_{coh}	Spreading factor $\frac{\tau_{\max}}{T_{\text{coh}}}$	Number of OFDM subcarriers N
$\alpha_1 = 0.003$	1.71 s	0.0077	507
$\alpha_2 = 0.03$	0.2 s	0.066	54
$\alpha_3 = 0.5$	0.0133 s	0.99	<1

Table 2.3: Summary of time variation of the long-range shallow-water channel (soft bottom $c_2 = 1550$ m/s).

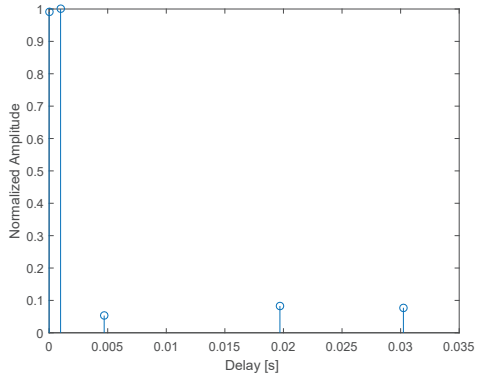
Long range shallow water (hard bottom)			
$f = 2$ kHz, $B = 300$ Hz, $\tau_{\max} = 0.343$ s, $L_{\max} = 105$			
	Coherence time T_{coh}	Spreading factor $\frac{\tau_{\max}}{T_{\text{coh}}}$	Number of OFDM subcarriers N
$\alpha_1 = 0.003$	$< T_s$	> 102.9	<1
$\alpha_2 = 0.03$	$< T_s$	> 102.9	<1
$\alpha_3 = 0.5$	$< T_s$	> 102.9	<1

Table 2.4: Summary of time variation of the long-range shallow-water channel (hard bottom $c_2 = 1650$ m/s).

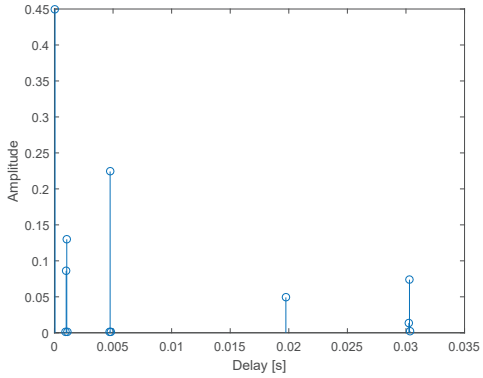
2.3.3 Short-Range Shallow-Water Channels

In the short-range shallow-water scenario, the channel depth is $d = 30$ m, the channel distance $l = 200$ m, the transmitter and receiver depths are $d_{\text{tx}} = 15$ m, $d_{\text{rx}} = 9.7$ m, respectively. We transmit signals with carrier frequency $f = 20$ kHz and bandwidth $B = 20$ kHz. Similar to Section 2.3.2, we simulated soft bottom and hard bottom cases in Figure 2.15. We can see that in this channel, there are not as many bounces as in Figure 2.14, for both soft and hard bottoms. Therefore the minimum delay $\tau_{\min}$ is relatively large, and also because the bandwidth here is 20 kHz, all the delays can be resolved.

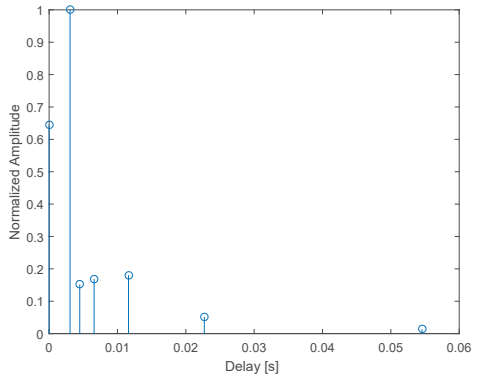
However, Table 2.6 indicates that this channel with a hard bottom is typically overspread because of its longer maximum delay. We choose the value of α corresponding to $f = 20$ kHz, but actually α will be larger, because if we implement OFDM in this broadband channel, the subcarrier frequency will be in a range of $[f - \frac{B}{2}, f + \frac{B}{2}] = [10\text{kHz}, 30\text{kHz}]$. For a narrowband channel, α can be approximated



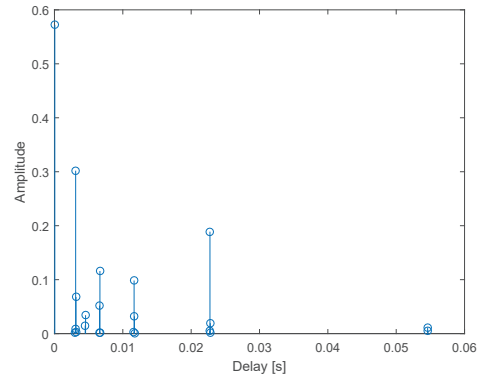
(a) Channel delay profile with soft bottom $c_2 = 1550$ m/s, $B_{\min} = 4$ kHz



(b) Channel delay profile after filtering effect with soft bottom $c_2 = 1550$ m/s



(c) Channel delay profile with hard bottom bottom $c_2 = 1650$ m/s, $B_{\min} = 2.8$ kHz



(d) Channel delay profile after filtering effect with hard bottom bottom $c_2 = 1650$ m/s

Figure 2.15: Different channel delay profiles in a shallow-water short-range scenario.

as a constant number over all the frequencies.

Short range shallow water (soft bottom)			
$f = 20$ kHz, $B = 20$ kHz, $\tau_{\max} = 30$ ms, $L_{\max} = 607$			
	Coherence time T_{coh}	Spreading factor $\frac{\tau_{\max}}{T_{\text{coh}}}$	Number of OFDM subcarriers N
$\alpha_1 = 0.1$	1s	0.3	19393
$\alpha_2 = 0.2$	20 ms	1.5	<1
$\alpha_3 = 2$	2.5 ms	12	<1
$\alpha_4 = 35$	0.2 ms	150	<1

Table 2.5: Summary of time variation of the short-range shallow-water channel (soft bottom $c_2 = 1550$ m/s).

Short range shallow water (hard bottom)			
$f = 20$ kHz, $B = 20$ kHz, $\tau_{\max} = 55$ ms, $L_{\max} = 1093$			
	Coherence time T_{coh}	Spreading factor $\frac{\tau_{\max}}{T_{\text{coh}}}$	Number of OFDM subcarriers N
$\alpha_2 = 0.2$	12 ms	4.58	<1
$\alpha_3 = 2$	2 ms	28	<1
$\alpha_4 = 35$	0.2 ms	280	<1

Table 2.6: Summary of time variation of the short-range shallow-water channel (hard bottom $c_2 = 1650$ m/s).

2.3.4 Meduim-Range Deep-Water Channels

In the medium-range deep-water scenario, the water depth is $d = 5$ km, the channel distance $l = 1$ km, and the depth of transmitter and receiver are up to 3 km. The carrier frequency and bandwidth are both 20 kHz. We put one transmitter at the depth $d_{\text{tx}} = 3$ km, and put 50 receivers vertically up to depth $d_{\text{rx}} = 3$ km, and then obtained delay profiles at each receiver (see Figure 2.16). We can see that this channel is an easy underspread channel because there is only a single path at each receiver location.

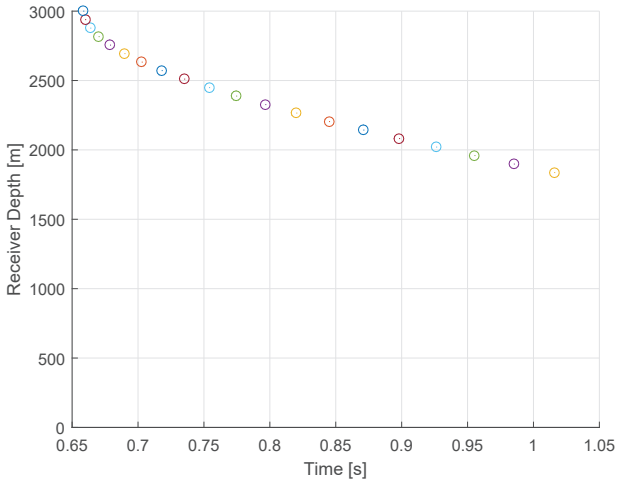


Figure 2.16: Different channel delay profiles at different receiver depths in a deep-water medium-range scenario.

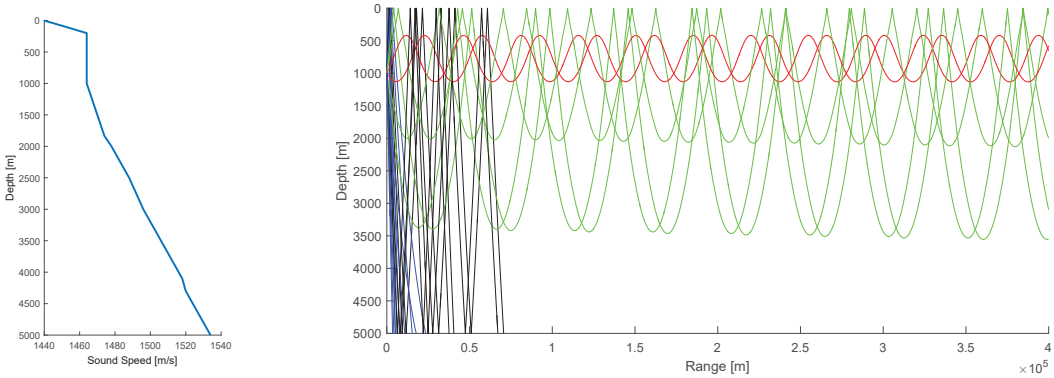
For $f = 20$ kHz, we choose the value of $\alpha = 0.16$, and the results of the time variation behavior is in shown Table 2.7.

Meduim range deep water			
$f = 20$ kHz, $B = 20$k Hz, $\tau_{\max} = 0$ ms, $L_{\max} = 1$			
	Coherence time T_{coh}	Spreading factor $\frac{\tau_{\max}}{T_{\text{coh}}}$	Number of OFDM sub-carriers N
$\alpha_1 = 0.16$	50 ms	0	1000

Table 2.7: Summary of time variation of medium-range deep-water channel.

2.3.5 Arctic Ocean Channels

Different from typical ocean channels in previous four scenarios, the sound speed in Arctic Ocean is generally increasing with the water depth (see Figure 2.17(a)), and it is only found in polar waters, slightly changing with the season [23]. The typical ray propagation model with this sound speed profile in shown in Figure 2.17(b) with beam degree ranging from $-50°$ to $50°$. The green curves represents the rays only hit the surface, the black curves have reflections on both surface and bottom, the blue curves only hit the bottom, and the red curves propagates with no boundaries. One interesting characteristic for Arctic Ocean is that the rays which only hit the surface or does not hit both boundaries can propagate in a long range.



(a) Typical sound speed profile in Arctic Ocean

(b) Typical ray propagation

Figure 2.17: Typical sound speed profile and ray propagation in Arctic Ocean (assuming flat surface and bottom without ice).

However, because of the ice cover on the surface and variation of the ocean floor,

the ray propagation model will be more complex than Figure 2.17(b), which will experience much more reflections and transmission loss, in turn, it will be more difficult for long range communications.

In following sections, we will apply the ice draft profile obtained in summer and a bathymetry file (see Figure 2 in [24]) to Bellhop software to generate the channel taps in deep water and shallow water respectively.

Deep-Water Channel in Arctic Ocean

In this scenario, the water depth is up to 4 km with the ocean floor applied, the depth of transmitter is $d_{\text{tx}} = 1000$ m, and the average of the thickness of the ice cover is 2.15 m. The carrier frequency is 250 Hz. The ray propagation model with 20 beams is in Figure 2.18.

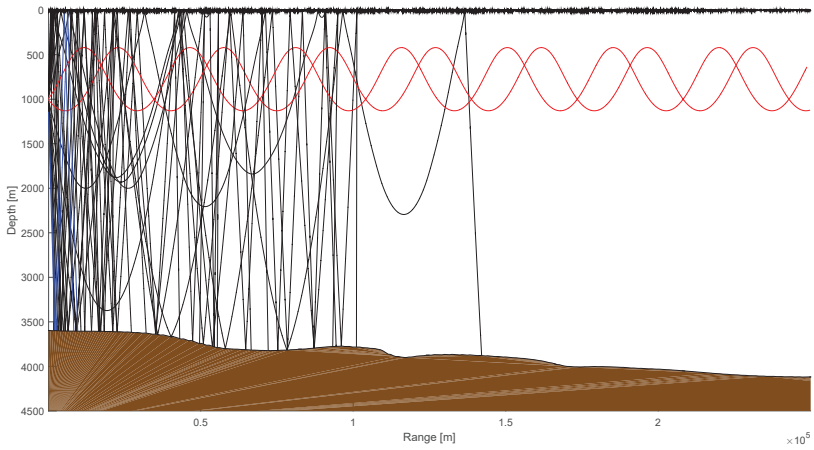
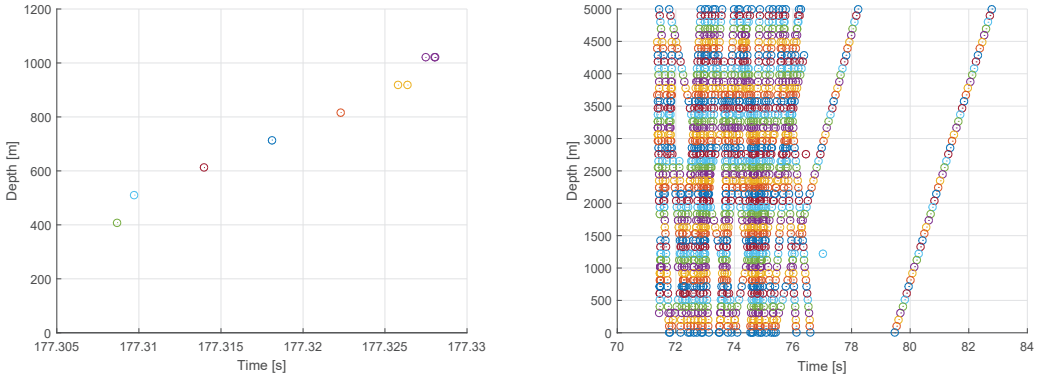


Figure 2.18: Ray propagation model in deep water with bathymetry and ice draft profile applied.

We can see that most of the rays are absorbed after 100 km because of the large number of reflections on the surface, except the rays without hitting the boundaries.

Then, we obtained the channel tap arrival profile at different receiver ranges in Figure 2.19. It is clear that at range 260 km the channel is sparse but with small minimum delay, and at range 104 km, the channel has larger numbers of delays and longer delay spread. An example of the delay-amplitude profile in Figure 2.19(b) is shown in Figure 2.20. This result is consistent with Figure 2.18.



(a) The channel tap arrival profile at 260 km (b) The channel tap arrival profile at 104 km

Figure 2.19: The channel tap arrival profiles at range 260 km and 104 km, $d_{\text{tx}} = 1000$ m, $f = 250$ Hz (deep water).

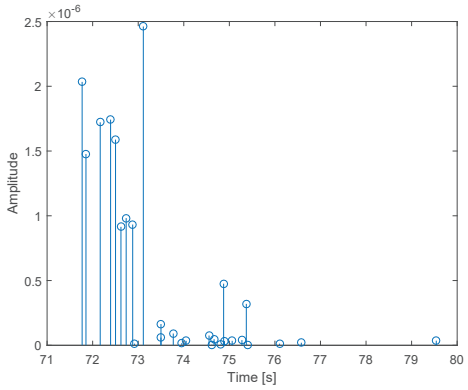


Figure 2.20: Channel delay-amplitude profile at range 104 km, $d_{\text{tx}} = 1000$ m, $d_{\text{rx}} = 102$ m, $f = 250$ Hz.

Shallow-Water Channel in Arctic Ocean

In shallow-water channel, the depth of the water is $d = 250$ m with assuming a flat and hard bottom, same ice draft profile applied as in deep-water scenario, and the depth of the transmitter is $d_{\text{tx}} = 100$ m. We choose this depth of the transmitter because in the experiment we find we can transmit signals at the longest range at this depth. The carries frequency is 250 Hz. Figure 2.21 is the ray propagation model.

From Figure 2.21, we can see that the longest distance the rays can propagate is around 20 km, which is much shorter than in deep water channel since in shallow water the signals will experience more reflections and attenuations. We present the channel tap arrival profiles in Figure 2.22. Different from the shallow water in other

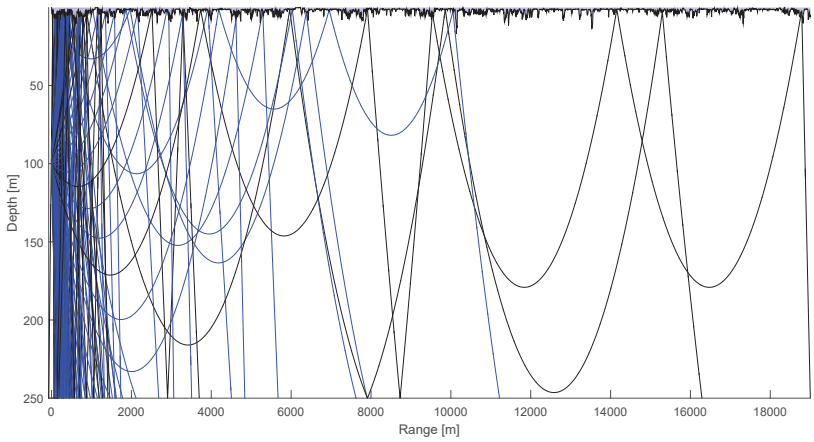
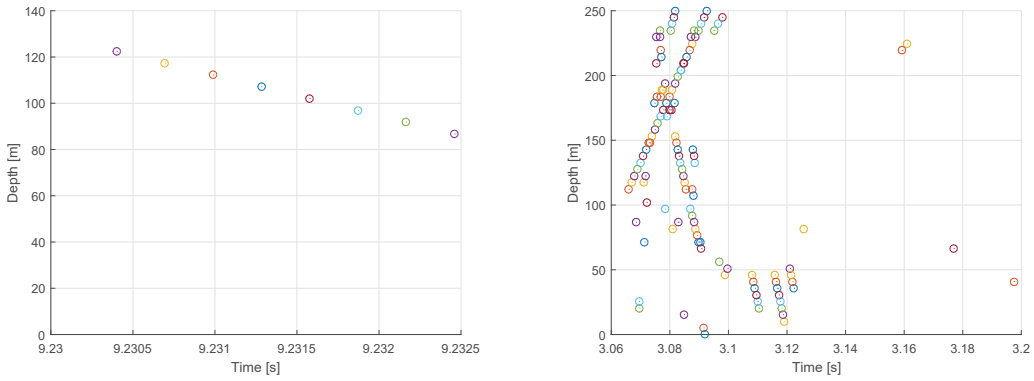


Figure 2.21: Ray propagation model in deep water with ice draft profile applied.

typical oceans, there are few bounces at a long range because the rough ice cover makes siginificant transmission loss.



(a) Channel tap arrival profile at 13 km (b) Channel tap arrival profile at 4.4 km

Figure 2.22: Channel tap arrival profiles at range 13 km and 4.4 km, $d_{\text{tx}} = 1000$ m, $f = 250$ Hz (shallow water).

2.3.6 Conclusion

From the experiments above, we can see that UWA channels have shown various sparsity levels. Deep water channels are usually sparse, but the shallow water channels with soft bottom are dense and hard to resolve. The channels can be overspread in a long range communications because of the long maximum delay. In general, the meduim range deep water channels are favorable channels to communicate.

Chapter 3

Sparse Channel Estimation

3.1 Introduction

Sparse channel estimation is an application of Compressed Sensing (CS). It is a novel technique to recover the sparse signal from fewer samples than the number of samples acquired from the Nyquist rate [25]. In a mathematical form, first consider a signal $\boldsymbol{x} \in \mathbb{C}^N$ having a sparse representation in a basis $\boldsymbol{\Psi} = [\psi_1, \psi_2, \ldots, \psi_N]$, where $\boldsymbol{\Psi}$ is a $N \times N$ matrix and $\psi_i, i = 1, 2, \ldots, N$ standing for the column of $\boldsymbol{\Psi}$. In this basis, $\boldsymbol{x} = \boldsymbol{\Psi\theta}$, where the coefficients θ_i are sparse. If there are k nonzero elements in vector $\boldsymbol{\theta}$, then we call the signal $\boldsymbol{x}$ as k-sparse signal in $\boldsymbol{\Psi}$ basis. For example, if $\boldsymbol{\Psi}$ is the discrete Fourier transform (DFT) matrix, then it means the frequency-domain signal has a sparse representation in the time domain. However, most of the time, a signal can be disturbed by some noise, therefore, the signal is called approximately k-sparse if it can be represented by most of the large coefficients in some basis up to a certain accuracy [26].

A compressed sensing problem is an under-determined problem for sparse signals. Consider we need to recover the signal $\boldsymbol{x}$ from the received samples $\boldsymbol{y}$, and the signal acquisition process can be written as

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{\Phi x} \\ &= \underbrace{\boldsymbol{\Phi\Psi}}_{\mathbf{A}} \boldsymbol{\theta} \end{aligned} \tag{3.1}$$

where $\boldsymbol{y}$ is a known vector of size $n \times 1$, $\boldsymbol{\theta}$ is an unknown vector of size $N \times 1$, and matrix $\mathbf{A}$ $(n \times N)$ is called sensing matrix or measurement matrix which we need to design. Based on the linear algebra, we need $n \geq N$ samples to have the exact answer of $\boldsymbol{\theta}$; otherwise, there will be multiple solutions. However, with the condition that $\boldsymbol{\theta}$ is sparse, then we can use fewer samples $(n < N)$ to recover the signal by l_1 minimization[25]

$$\min ||\boldsymbol{\theta}||_{l_1}, \text{ subject to } \boldsymbol{y} = \mathbf{A}\boldsymbol{\theta} \tag{3.2}$$

where $||\boldsymbol{\theta}||_{l_1} = \sum_{i=1}^{N} \theta_i$. If the received samples is interfered with noise $\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{w}$, where $\boldsymbol{w}$ is assumed to be white Gaussian noise with zero mean of size $n \times 1$, we recover the signal by solving an alternative of problem 3.2

$$\min ||\boldsymbol{\theta}||_{l_1}, \text{ subject to } ||\boldsymbol{y} - \mathbf{A}\boldsymbol{\theta}||_{l_2} \leq \epsilon \tag{3.3}$$

where ϵ is a small number indicating the variance of the noise, and for arbitrary vector $\boldsymbol{z}$, the l_2 norm is $||\boldsymbol{z}||_{l_2} = (\sum_i z_i^2)^{\frac{1}{2}}$.

From the literature [26, 27], the l_1 minimization problem 3.2 is also called Basis Pursuit (BP) and its alternative problem 3.3 is called Basis Pursuit De-Noising (BPDN) or LASSO. The use of l_1 norm constraint to solve sparsity problem has a long history since 1930s [28, 29, 30]. It is a natural mathematical choice to measure the sparsity, compared to the l_0 constraint (the number of nonzero elements in a sparse vector) which is non-tractable and l_2 constraint does not require the sparsity [31]. Moreover, based on the paper [32], for most of the under-determined system, the solution of l_1 minimization problem is the exact sparse answer.

Minimizing l_1 norm subject to a constraint can be solved by the linear programming; however, it has a disadvantage of computational complexity. Other ways to recover the sparse signal include the greedy algorithm (OMP) and iterative thresholding. For OMP, assume we know that the signal is k-sparse, then the algorithm can stop the iterations at kth step by choosing one nonzero element at each step. However in reality we have no knowledge about the value of k; therefore we usually choose different stop criteria for each instance. OMP has the advantage of its speed, and also in paper [8], it has been rigorously proved that OMP can recover the exact k-sparse signal with measurements nearly proportional to k. For iterative thresholding, at each step it simply sets a threshold to choose the subset of columns of sensing matrix which is highly correlated to the received signals. However, even though iterative thresholding is computationally efficient, it has worse performance compared with OMP and l_1 minimization method [33]. Another algorithm with both advantages of low computational complexity and competitive performance is AMP. It is derived from the sum product belief algorithm and has a crucial term added to the iterative thresholding formula which influences the performance significantly.

For all the algorithms to solve the CS problems efficiently, the sensing matrix $\mathbf{A}$ is required to show randomness. This randomness is described by the restricted

isometry property (RIP) [25].

Definition 1. *A matrix* $\mathbf{A}$ *satisfied the restricted isometry property (RIP) of order* k *if there exists a* $\delta_k \in (0,1)$ *such that*

$$(1-\delta_k)||\boldsymbol{\theta}||_{l_2}^2 \leq ||\mathbf{A}\boldsymbol{x}||_{l_2}^2 \leq (1+\delta_k)||\boldsymbol{x}||_{l_2}^2 \tag{3.4}$$

holds for all $\boldsymbol{x}$.

This property means that matrix $\mathbf{A}$ can approximately preserve the Euclidean length of k-sparse signals. To construct such matrices holding RIP with highly probability, we prefer the random matrix in which each element is independent and identically distributed (i.i.d.) with the number of rows $n \geq Ck\log(N/k)$, where C is a constant depending on each scenario [25]. To measure the estimation capability of the matrix, we calculate the incoherence of the matrix defined as the maximum cross-correlation of any two columns [19].

$$\mu \triangleq \max_{l \neq l'} < \boldsymbol{A}_l, \boldsymbol{A}_l' > \tag{3.5}$$

where l, l' are both column indexes, and $< \boldsymbol{A}_l, \boldsymbol{A}_l' >$ represents the inner product of two columns and $\mu \in (0,1)$. The incoherence of the matrix is related to how many nonzero elements (s) that we can recover shown in Equation 3.6 [19]. So the less μ is, the less sparse signal we can handle.

$$\mu < \frac{1}{2s-1} \tag{3.6}$$

In this Chapter, we consider the CS problem in the sparse UWA channel scenario where the sparse signal is a channel tap vector denoted by $\boldsymbol{h}$, and it is sparse in its acquired delay domain. Therefore the basis $\boldsymbol{\Psi}$ in Equation 3.1 is an identity matrix. The sensing matrix $\mathbf{A}$ is constructed by shifted pilot signals such that each row is one-shifted vector of the previous row. To satisfy the requirement for the randomness, the pilot signals are designed to be shift-orthogonal or with low autocorrelations. The problem setup details will be described in Section 3.4.

The performances of sparse algorithms are measured by the phase transition curve, which represents the trade-off between the sparsity and under-sampling. To be specific, consider the CS problem in Equation 3.1, then the sparsity level of the signal is denoted by

$$\rho = \frac{k}{n} \tag{3.7}$$

where k is the number of nonzero elements of the signal vector or the approximate number of nonzero elements if we have noise. The under-sampling level is denoted by

$$\delta = \frac{n}{N} \tag{3.8}$$

Therefore the phase transition curve is a $\rho - \delta$ curve. The larger δ, the large ρ we can tolerate for the success of the algorithms. In other words, the more measurements we take, the more nonzero elements in the signals we can recover. For testing the success of the algorithms, we calculate the MSE between the estimated signal vector and the original one. If we denote the estimation as $\hat{\boldsymbol{h}}$, then the MSE is expressed as

$$\text{MSE} = \mathbb{E}\{||\hat{\boldsymbol{h}} - \boldsymbol{h}||^2\} \tag{3.9}$$

If the MSE is smaller than a certain threshold, then it means that the algorithm recovers $\boldsymbol{h}$ successfully.

The following two sections 3.2 and 3.3 will focus on the details of two algorithms (OMP and AMP) that solve the CS problem. To adjust the algorithms to the sparse channel estimation problem we consider, some parameters and steps need to be tuned appropriately. Section 3.4 will describe the problem setup and present the numerical results of these two algorithms.

3.2 Orthogonal Matching Pursuit (OMP)

Consider the problem setup in Equation 3.1 with random matrix $\mathbf{A}$, and we try to use the OMP algorithm to find the sparsest solution of $\boldsymbol{\theta}$. The estimated result is denoted by $\hat{\boldsymbol{\theta}}$. The first step is to initialize the residual error to be equal to the measurement vector.

$$\boldsymbol{r}_0 = \boldsymbol{y} \tag{3.10}$$

Next at each iteration t, we find the column in $\mathbf{A}$ that has the maximum correlation with residual $\boldsymbol{r}_{t-1}$ in the previous iteration, and then put the column index s_t into the index set $\boldsymbol{S}_{t-1}$. The column index s_t at the tth iteration is computed as

$$s_t = \arg \max_{l=1,\ldots,N} < \boldsymbol{r}_{t-1}, \boldsymbol{A}_l > \tag{3.11}$$

This step simply calculate the inner product between residual $\boldsymbol{r}_t$ and each column of $\mathbf{A}$ and then choose the column with largest result. Some variations of OMP exclude

the chosen columns in this step, which means each column of $\mathbf{A}$ can be chosen only once, however, we follow the procedures in paper [8]. Then the index set $\boldsymbol{S}_t$ becomes

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-1} \cup \{s_t\} \tag{3.12}$$

The estimated result $\hat{\boldsymbol{\theta}}$ is updated at tth iteration by solving a least square problem using the columns in $\mathbf{A}$ which have been already chosen

$$\begin{aligned} \hat{\boldsymbol{\theta}}_t &= \arg\min_{\boldsymbol{\theta}} ||\boldsymbol{y} - \mathbf{A}_{\boldsymbol{S}_t}\boldsymbol{\theta}||_{l_2}^2 \\ &= (\mathbf{A}_{\boldsymbol{S}_t}^h \mathbf{A}_{\boldsymbol{S}_t})^{-1} \mathbf{A}_{\boldsymbol{S}_t}^h \boldsymbol{y} \end{aligned} \tag{3.13}$$

where $\mathbf{A}_{\boldsymbol{S}_t}^h$ denotes the Hermitian transpose of matrix $\mathbf{A}_{\boldsymbol{S}_t}$. The next step is to calculate the new residual error

$$\boldsymbol{r}_t = \boldsymbol{y} - \mathbf{A}_{\boldsymbol{S}_t}\hat{\boldsymbol{\theta}} \tag{3.14}$$

We then recursively iterates following the step from Equation 3.11 to Equation 3.14 until the iteration index t reaches k, the exact number of non-zero elements in vector $\boldsymbol{\theta}$. However, in reality, we do not know about k or $\boldsymbol{\theta}$ is approximately sparse, therefore, for the approximate sparse signal, we use relatively thresholding (RT) criteria to stop the iteration [19].

$$10\log_{10}\frac{\max\{\hat{\theta_1}, \ldots, \hat{\theta_N}\}}{\min\{\hat{\theta_1}, \ldots, \hat{\theta_N}\}} < \gamma_1 \tag{3.15}$$

Because OMP chooses the significant elements in $\hat{\boldsymbol{\theta}}$ in a amplitude-decreasing order, here we set γ_1 in dB to determine the stop iteration for comparing the maximum element and the minimum element in $\hat{\boldsymbol{\theta}}$. For the exactly sparse signal, we use the residual decay rate (RDR) stopping criteria expressed as

$$\log_{10}\frac{||\boldsymbol{r}_{t-1}||_{l2}}{||\boldsymbol{r}_t||_{l2}} > \gamma_2 \tag{3.16}$$

It is because in the experiments we find that the residual will suddenly drop after kth iteration. We did not simply use the amplitude of the residuals to determine the stopping criteria since OMP would pick the noise tap after kth iteration to make the residual smaller. The RDR stopping criteria is implemented in Section 3.4.3.

If there is noise vector $\boldsymbol{w}$ (AWGN with variance N_0) added in the measurement vector $\boldsymbol{y}$, then the measurement formula is expressed as

$$\boldsymbol{y} = \mathbf{A}\boldsymbol{\theta} + \boldsymbol{w} \tag{3.17}$$

and the estimate update step in Equation 3.13 should be changed as

$$\hat{\boldsymbol{\theta}}_t = (\mathbf{A}^h_{\boldsymbol{S}_t}\mathbf{A}_{\boldsymbol{S}_t} + N_0\mathbf{I})^{-1}\mathbf{A}^h_{\boldsymbol{S}_t}\boldsymbol{y} \tag{3.18}$$

where $\mathbf{I}$ is an identity matrix with size $n \times n$. In summary, the procedures of OMP is in Algorithm 1.

Algorithm 1 Orthogonal Matching Pursuit (OMP)

Input:

$n \times 1$ measurement vector $\boldsymbol{y}$

$n \times N$ random sensing matrix $\mathbf{A}$

Noise variance N_0

Relative thresholding γ in dB

Output:

An estimate $n \times 1$ vector $\hat{\boldsymbol{\theta}}$

1: Initialization: $\boldsymbol{r_0} = \boldsymbol{y}$, $\boldsymbol{S}_0 = \emptyset$, $t = 1$

2: **while** $10\log_{10}\frac{\max\{\hat{\theta_1},\ldots,\hat{\theta_N}\}}{\min\{\hat{\theta_1},\ldots,\hat{\theta_N}\}} < \gamma_1$ (approximate sparse) or $\log_{10}\frac{||\boldsymbol{r}_{t-1}||_{l2}}{||\boldsymbol{r}_t||_{l2}} > \gamma_2$ (exactly sparse) **do**

3: $s_t = \arg\max_{l=1,\ldots,N} < \boldsymbol{r}_{t-1}, \boldsymbol{A}_l >$

4: $\boldsymbol{S}_t = \boldsymbol{S}_{t-1} \cup \{s_t\}$

5: $\hat{\boldsymbol{\theta}}_t = (\mathbf{A}^h_{\boldsymbol{S}_t}\mathbf{A}_{\boldsymbol{S}_t} + N_0\mathbf{I})^{-1}\mathbf{A}^h_{\boldsymbol{S}_t}\boldsymbol{y}$

6: $\boldsymbol{r}_t = \boldsymbol{y} - \mathbf{A}_{\boldsymbol{S}_t}\hat{\boldsymbol{\theta}}$

7: $t = t + 1$

8: **end while**

3.3 Approximate Message Passing (AMP)

AMP algorithm is derived from the graphical model theory and message passing algorithm [27]. It simplifies the procedures of message passing that requires tracking of $2nN$ messages. The following sections will first look at the derivation from message passing to AMP briefly and then describe the AMP algorithm for both real and complex numbers.

3.3.1 From Message Passing to AMP

Consider the problem in Equation 3.1, $\boldsymbol{y} = \mathbf{A}\boldsymbol{\theta}$, and denote the residual vector as $\boldsymbol{z}$. Then the factor graph for message passing between N variable nodes $\theta_1, \ldots, \theta_N$ and n factor nodes $z_1, \ldots, z_n$ is illustrated in Figure 3.1. In this figure, $i \in \{1, 2, \ldots, N\}$ is the index of variable nodes and $a \in \{1, 2, \ldots, n\}$ is the index of factor nodes. $\hat{\theta}_{i\to a}$ represents the message passing from the ith variable node to the ath factor node, and similarly $z_{a\to i}$ represents the message passing from the ath factor node to the ith variable node.

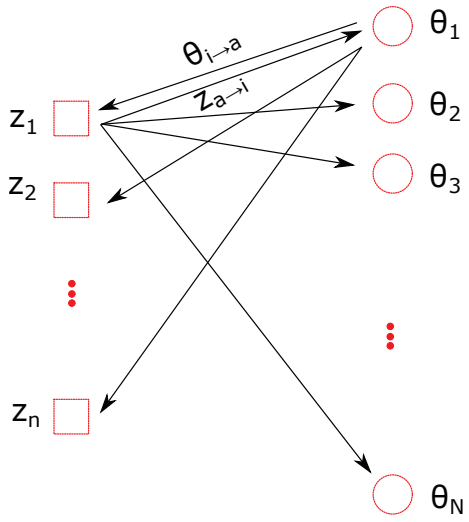


Figure 3.1: Factor graph of message passing algorithm.

Each factor node will pass message to every variable node through the edge and vice versa. We follow the derivation in the reference [27]. The requirement for matrix $\mathbf{A}$ is that each entry is i.i.d with zero mean and variance $1/n$, and the l_2 norm of each column is normalized to be 1, i.e. $||\boldsymbol{A}_l||_{l_2}^2 = 1$, where l is the column index. The message passing rules are expressed as

$$\hat{\theta}_{i\to a}^{t+1} = \eta(\sum_{b\neq a} A_{bi} z_{b\to i}^t; \hat{\tau}^t) \tag{3.19}$$

$$z_{a\to i}^t = y_a - \sum_{j\neq i} A_{aj} \hat{\theta}_{j\to a}^t \tag{3.20}$$

$$\hat{\tau}^{t+1} = \frac{\hat{\tau}^t}{n} \sum_{j\in[N]} \eta'(\sum_{b\in[n]} A_{bj} z_{b\to j}^t; \hat{\tau}^t) \tag{3.21}$$

where t is the iteration number and A_{ai} denotes the element in ath row and ith column. $\eta(x;b)$ is a thresholding function applied component-wise expressed as

$$\eta(x;b) = \begin{cases} x-b, & \text{if } b < x \\ 0, & \text{if} -b \leq x \leq b \\ x+b, & \text{if } x < -b \end{cases} \tag{3.22}$$

and $\eta(\cdot)' = \frac{\partial \eta(\cdot)}{\partial x}$ is also a component-wise function. At each iteration, the message received at each variable node from ith factor node is a sum of the messages passing from all the other variable nodes to the ith factor node. The purpose of the thresholding function $\eta(\cdot)$ is to force the sparsity of the message $\hat{\theta}_{i\to a}$ with thresholding level $\hat{\tau}^t$ updated at each iteration. In a heuristic point of view, $\hat{\tau}^t$ is like the variance of $z_{a\to i}$ and we will see later that $\hat{\tau}^t$ can be tuned using the variance of the residual vector $\boldsymbol{z}$.

Observing the Equation 3.19, it is easy to see that the right-hand term depends weakly on the index a, and the right-hand side of Equation 3.20 also depends weakly on the index i. To construct the AMP algorithm, we first assume that there exists $\triangle\theta^t_{i\to a}, \triangle z^t_{a\to i} = O(\frac{1}{\sqrt{N}})$ such that

$$\hat{\theta}^t_{i\to a} = \hat{\theta}^t_i + \triangle\hat{\theta}^t_{i\to a} + O(1/N) \tag{3.23}$$

$$z^t_{a\to i} = z^t_a + \triangle z^t_{a\to i} + O(1/N), \quad \text{for all } i, a \tag{3.24}$$

where $\triangle\hat{\theta}^t_{i\to a}, \triangle z^t_{a\to i} = O(\frac{1}{\sqrt{N}})$ are the errors that depend on the choice of the edge. Substituting Equations 3.23 and 3.24 into the equations of message passing rules (3.19, 3.20, 3.21), we obtain

$$\begin{aligned} \hat{\theta}^{t+1}_{i\to a} &= \eta(\sum_{b\neq a} A_{bi} z^t_{b\to i}) \\ &= \eta(\sum_{b\in[n]} A_{bi} z^t_{b\to i} - A_{ai} z^t_{a\to i}) \\ &= \eta(\sum_{b\in[n]} A_{bi} z^t_b + \sum_{b\in[n]} A_{bi}\triangle z^t_{b\to i} - A_{ai} z^t_a) + O(\frac{1}{N}) \\ &= \underbrace{\eta(\sum_{b\in[n]} A_{bi} z^t_b + \sum_{b\in[n]} A_{bi}\triangle z^t_{b\to i})}_{\hat{\theta}^{t+1}_i} \underbrace{- A_{ai} z^t_a \eta'(\sum_{b\in[n]} A_{bi} z^t_b + \sum_{b\in[n]} A_{bi}\triangle z^t_{b\to i})}_{\triangle\hat{\theta}^{t+1}_{i\to a}} \qquad (3.25) \\ &\quad + O(\frac{1}{N}) \qquad (3.26) \end{aligned}$$

where the last two step has applied the Taylor expansion, and the thresholding parameter τ^t is omitted for convenience. Similarly, for $z^t_{a\to i}$, we have

$$\begin{aligned} z^t_{a\to i} &= y_a - \sum_{j\neq i} A_{aj}\hat{\theta}^t_{j\to a} \\ &= y_a - \sum_{j\in[N]} A_{aj}\hat{\theta}^t_{j\to a} + A_{ai}\hat{\theta}^t_{i\to a} \\ &= \underbrace{y_a - \sum_{j\in[N]} A_{aj}\hat{\theta}^t_j - \sum_{j\in[N]} A_{aj}\triangle\hat{\theta}^t_{j\to a}}_{z^t_a} + \underbrace{A_{ai}\hat{\theta}^t_{i\to a}}_{\triangle z^t_{a\to i}} + O(\frac{1}{N}) \end{aligned} \tag{3.27}$$

In underbrace, the updates of the variable nodes and factor nodes are defined as

$$\begin{aligned} \hat{\theta}^{t+1}_i &= \eta(\sum_{b\in[n]} A_{bi}z^t_b + \sum_{b\in[n]} A_{bi}\triangle z^t_{b\to i}) \\ &= \eta(\sum_{b\in[n]} A_{bi}z^t_b + \sum_{b\in[n]} A^2_{bi}\hat{\theta}^t_i) \\ &= \eta(\sum_{b\in[n]} A_{bi}z^t_b + \hat{\theta}^t_i) \end{aligned} \tag{3.28}$$

$$\begin{aligned} z^t_a &= y_a - \sum_{j\in[N]} A_{aj}\hat{\theta}^t_j + \sum_{j\in[N]} A_{aj}\triangle\hat{\theta}^t_{j\to a} \\ &= y_a - \sum_{j\in[N]} A_{aj}\hat{\theta}^t_j + \sum_{j\in[N]} A^2_{aj}z^{t-1}_a\eta'(\sum_{b\in[n]} A_{bi}z^{t-1}_b + \sum_{b\in[n]} A_{bi}\triangle z^{t-1}_{b\to i}) \\ &= y_a - \sum_{j\in[N]} A_{aj}\hat{\theta}^t_j + \sum_{j\in[N]} A^2_{aj}z^{t-1}_a\eta'(\sum_{b\in[n]} A_{bi}z^{t-1}_b + \hat{\theta}^{t-1}_j) \\ &= y_a - \sum_{j\in[N]} A_{aj}\hat{\theta}^t_j + \frac{1}{n}\sum_{j\in[N]} z^{t-1}_a\eta'(\sum_{b\in[n]} A_{bi}z^{t-1}_b + \hat{\theta}^{t-1}_j) \\ &= y_a - \sum_{j\in[N]} A_{aj}\hat{\theta}^t_j + \frac{1}{\delta}z^{t-1}_a < \eta'(\sum_{b\in[n]} A_{bi}z^{t-1}_b + \hat{\theta}^{t-1}_j) > \end{aligned} \tag{3.29}$$

where $< \eta'(\cdot) >$ denotes the mean of vector $\eta'(\cdot)$, and $\delta = \frac{n}{N}$. For update of $\hat{\tau}^t$, we have

$$\begin{aligned} \hat{\tau}^{t+1} &= \frac{\hat{\tau}^t}{n}\sum_{j\in[N]} \eta'(\sum_{b\in[n]} A_{bj}z^t_{b\to j}; \hat{\tau}^t) \\ &= \frac{\hat{\tau}^t}{n}\sum_{j\in[N]} \eta'(\sum_{b\in[n]} A_{bi}z^t_b + \sum_{b\in[n]} A_{bi}\Delta z^t_{b\to i}; \hat{\tau}^t) \\ &= \frac{\hat{\tau}^t}{\delta} < \eta'(\sum_{b\in[n]} A_{bi}z^t_b + \hat{\theta}^t_j; \hat{\tau}^t) > \end{aligned} \tag{3.30}$$

Here, the derivation of last two steps are the same as the derivation in Equation 3.29. For the measurement vector with AWGN noise, $\boldsymbol{y} = \mathbf{A}\boldsymbol{\theta} + \boldsymbol{w}$, with noise variance N_0, $\hat{\tau}^t$ is tuned as

$$\hat{\tau}^{t+1} = \sqrt{N_0 + \left(\frac{\hat{\tau}^t}{\delta} < \eta'(\sum_{b \in [n]} A_{bi} z_b^t + \hat{\theta}_j^t; \hat{\tau}^t) > \right)^2} \tag{3.31}$$

Now the derivation of AMP is finished and the thresholding parameter $\hat{\tau}^t$ is automatically tuned in this version. The summary of AMP in vector form is illustrated in Algorithm 2.

Algorithm 2 Approximate Message Passing (AMP)

Input:

$n \times 1$ measurement vector $\boldsymbol{y}$

$n \times N$ random sensing matrix $\mathbf{A}$ with each entry $\mathbb{E}(A_{ij}) = 0, \mathbb{E}(A_{ij}^2) = \frac{1}{n}$, and each column $||\boldsymbol{A}_l||_{l_2}^2 = 1$, for $i = 1, \ldots, n$, $j = 1, \ldots, N$,$l = 1, \ldots, N$

Noise variance N_0

Iteration number I

Output:

An estimate $n \times 1$ vector $\hat{\boldsymbol{\theta}}$

1: Initialization: $\boldsymbol{z}^0 = \boldsymbol{y}$, $\boldsymbol{\theta}^0 = \mathbf{0}$, $t = 0$

2: **while** $t < I$ **do**

3: $\quad \hat{\boldsymbol{\theta}}^{t+1} = \eta(\mathbf{A}^T \boldsymbol{z}^t + \boldsymbol{\theta}^t; \hat{\tau}^t)$

4: $\quad \boldsymbol{z}^t = \boldsymbol{y} - \mathbf{A}\hat{\boldsymbol{\theta}}^t + \frac{1}{\delta}\boldsymbol{z}^{t-1} < \eta'(\mathbf{A}^T \boldsymbol{z}^{t-1} + \hat{\boldsymbol{\theta}}^{t-1}; \hat{\tau}^{t-1}) >$

5: $\quad t = t + 1$

6: $\quad \hat{\tau}^{t+1} = \sqrt{N_0 + \left(\frac{\hat{\tau}^{t-1}}{\delta} < \eta'(\mathbf{A}^T \boldsymbol{z}^{t-1} + \hat{\boldsymbol{\theta}}^{t-1}; \hat{\tau}^{t-1}) > \right)^2}$

7: **end while**

Compared with regular iterative thresholding algorithms (see [34]), the AMP has a crucial term, $\frac{1}{\delta}\boldsymbol{z}^{t-1} < \eta'(\mathbf{A}^T \boldsymbol{z}^{t-1} + \hat{\boldsymbol{\theta}}^{t-1}; \hat{\tau}^{t-1}) >$, added to the residual $\boldsymbol{z}^t$, which makes it outperform the iterative thresholding. The expression $\eta'(\cdot)$ actually computes the number of nonzero elements in the previous estimate, therefore, alternatively $\boldsymbol{z}^t$ can be expressed as

$$\boldsymbol{z}^t = \boldsymbol{y} - \mathbf{A}\hat{\boldsymbol{\theta}}^t + \frac{||\hat{\boldsymbol{\theta}}^t||_0}{n} \boldsymbol{z}^{t-1} \tag{3.32}$$

where the zero norm of $\hat{\boldsymbol{\theta}}^t$ is the number of nonzeros in this vector.

An important note is that the thresholding parameter $\hat{\tau}^t$ is a scaler and it sets the same threshold to each entry in $\hat{\boldsymbol{\theta}}^t$. As iteration goes, the MSE will converge to a nearly zero value if the algorithm works successfully. The MSE threshold can be set for each problem setup to determine the success of the algorithm and we can also calculate the success rate for running Monte Carlo simulations. The iteration number I is chosen based on the convergence speed in each scenario, and usually $I = 300$ is enough for a successful sparse signal recovery. More parameter settings will be articulate in Section 3.4.

3.3.2 Complex Approximate Message Passing (CAMP)

The AMP algorithm in previous section is derived for real numbers. For complex case, we follow the similar the derivation in reference [10] but have the automatic updated thresholding parameter $\hat{\tau}^t$. The summary is in Algorithm 3.

Here, $R(\cdot)$ and $I(\cdot)$ represent the real and imaginary part of a complex number respectively, $\mathbf{A}^h$ is the Hermitian transpose of $\mathbf{A}$, and A^*_{aj} is the conjugate of element A_{aj}. The thresholding function $\eta(\cdot)$ for complex numbers is defined as

$$\eta(u+iv;\tau) = \begin{cases} u+iv-\frac{\tau(u+iv)}{\sqrt{u^2+v^2}}, & \text{if } \sqrt{u^2+v^2} > \tau \\ 0, & \text{otherwise} \end{cases} \tag{3.33}$$

and $\frac{\partial \eta^R}{\partial y}, \frac{\partial \eta^R}{\partial y}$ and $\frac{\partial \eta^I}{\partial y}, \frac{\partial \eta^I}{\partial y}$ denote the partial derivatives of η^R and $\eta^I$with respect to the real and imaginary parts of the input.

3.3.3 Alternative Tuning of Thresholding Parameter and Stop Criteria

In the original papers of AMP and CAMP (see [9, 10]), there is a free paramter λ in the expression of the thresholding parameter $\hat{\tau}^t$ and need to be tuned for each scenario.

$$\hat{\tau}^t = \lambda\sqrt{(\hat{\sigma}^t)^2 + N_0} \tag{3.34}$$

Here, $\hat{\sigma}^t$ is called formal MSE and is the prediction of the true MSE at each iteration. Its recursive expression is

$$\left(\hat{\sigma}^{t+1}\right)^2 = \mathbb{E}\left\{\left[\eta\left(X + \sqrt{\frac{\left(\hat{\sigma}^t\right)^2}{\delta} + N_0}Z; \hat{\tau}^t\right) - X\right]^2\right\} \quad \text{(for AMP)} \tag{3.35}$$

$$\left(\hat{\sigma}^{t+1}\right)^2 = \mathbb{E}\left\{\left[\eta\left(X + \sqrt{\frac{\left(\hat{\sigma}^t\right)^2}{\delta} + N_0}Z_1 + i\sqrt{\frac{\left(\hat{\sigma}^t\right)^2}{\delta} + N_0}Z_2; \hat{\tau}^t\right) - X\right]^2\right\} \quad \text{(for CAMP)} \tag{3.36}$$

where X has the same distribution as the input vector $\boldsymbol{\theta}$, and $Z \sim \mathcal{N}(0,1)$, $Z_1, Z_2 \sim \mathcal{N}(0, \frac{1}{2})$. The analysis of formal MSE is called State Evolution (SE), and it can theoretically track the performance of AMP and CAMP and determine the success region ($\delta - \rho$ phase transition curve). This has been rigorously proved for $\mathbf{A}$ being Gaussian matrix [35]. Therefore, if we know the distribution of $\boldsymbol{\theta}$ and have the Gaussian sensing matrix $\mathbf{A}$, we can apply this approach to update $\hat{\tau}^t$. The free parameter λ need to be tuned such that the algorithms achieve the highest phase transition curve.

However, different from the thresholding updating steps in Algorithm 2 and 3, this approach does not update $\hat{\tau}^t$ automatically and it needs extensive experiments to find the optimal solution of λ. Moreover, in practice, we do not know the distribution of $\boldsymbol{\theta}$ beforehand and sometimes $\mathbf{A}$ is not ideally Gaussian. An alternative way to update $\hat{\tau}^t$ is to apply the variance of the residual $\boldsymbol{z}^t$ but still need free parameter λ [36]. The expression is

$$\hat{\tau}^t = \lambda\sqrt{\frac{||\boldsymbol{z}^t||_{l_2}^2}{n}} \tag{3.37}$$

Inspired by this approach, we propose another way to update $\hat{\tau}^t$ automatically by minimizing $||\boldsymbol{z}^t||_{l_2}^2$ over $\hat{\tau}^t$, that is

$$\hat{\tau}^t = \arg\min_{\hat{\tau}^t \in [0, \hat{\tau}^{t-1}]} ||\boldsymbol{z}^t||_{l_2}^2 \tag{3.38}$$

where $\hat{\tau}^t$ is getting smaller as the estimated sparsity level getting higher and closer to the original signal. The advantage of this approach is that the algorithm will not diverge, compared with Algorithm 2 and 3 having small probability of diverging

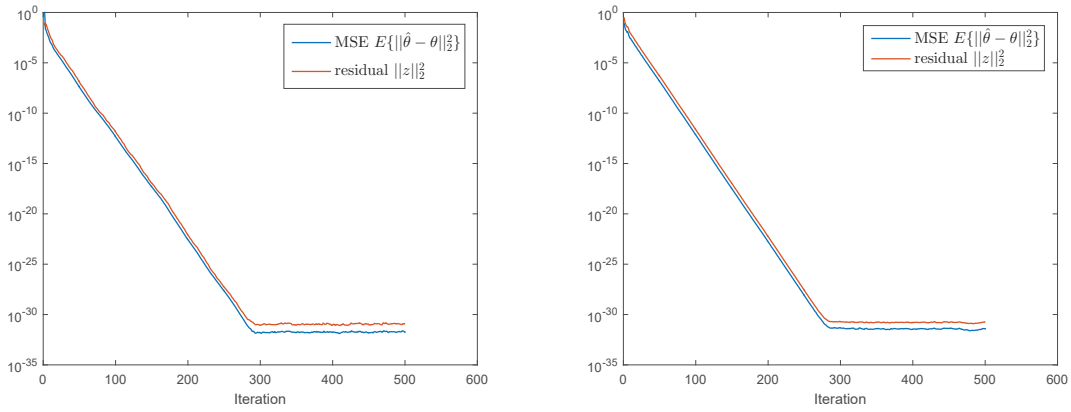
even in the success region. However, the computation complexity is the main issue especially for complex numbers with large n and N.

Since the iteration speed of CAMP is much slower than AMP, an constant iteration number I which is chosen by trial and error would not be the best choice for stopping the iterations. Alternatively, in the converging case, we compute the variance of the squared l_2 norm of the residuals $||\boldsymbol{z}||_{l_2}^2$ in the last six iterations, and if it is smaller than some small number ($< 10^{-5}$) ϵ_1, then we stop the algorithms. For the diverging case, if $||\boldsymbol{z}||_{l_2}^2$ is larger than some large number ($10^0 \sim 10^1$) ϵ_2, then we stop. In formula, the stopping criteria is

$$var(||\boldsymbol{z}^t||_{l_2}^2, ||\boldsymbol{z}^{t-1}||_{l_2}^2, \ldots, ||\boldsymbol{z}^{t-5}||_{l_2}^2) \quad < \quad \epsilon_1 \quad \text{,or} \tag{3.39}$$

$$||\boldsymbol{z}^t||_{l_2}^2 \quad > \quad \epsilon_2 \tag{3.40}$$

The reason for doing this is that the MSE of AMP and CAMP will converge to a nearly constant value if the algorithms successfully recover the signal (see Figure 3.2), and the residual will also stay nearly unchanged. Therefore, we can compute the variance of the squared l_2 norm of the residuals in the last several iterations to determine the algorithms converged or not.



(a) MSE map of AMP

(b) MSE map of CAMP

Figure 3.2: An example of MSE map of AMP algorithm and in the converging case. Blue curve is the MSE curve and red curve is the squared l_2 norm of the residuals. They both converge at almost the same iteration. For both AMP and CAMP, $\mathbf{A}$ is Gaussian matrix, $N = 800, \delta = 0.3, \rho = 0.2$, and the nonzero entries of $\boldsymbol{\theta}$ is Rayleigh distributed.

Algorithm 3 Complex Approximate Message Passing (CAMP)

Input:

$n \times 1$ measurement vector $\boldsymbol{y}$

$n \times N$ random sensing matrix $\mathbf{A}$ with each entry $\mathbb{E}(A_{ij}) = 0, \mathbb{E}(A_{ij}^2) = \frac{1}{n}$, and each column $||\boldsymbol{A}_l||_{l_2}^2 = 1$, for $i = 1, \ldots, n$, $j = 1, \ldots, N$, $l = 1, \ldots, N$

Noise variance N_0

Iteration number I

Output:

An estimate $n \times 1$ vector $\hat{\boldsymbol{\theta}}$

Initialization: $\boldsymbol{z}^0 = \boldsymbol{y}$, $\boldsymbol{\theta}^0 = \mathbf{0}$, $t = 0$

while $t < I$ **do**

$\hat{\boldsymbol{\theta}}^{t+1} = \eta\left(\mathbf{A}^h \boldsymbol{z}^t + \boldsymbol{\theta}^t; \hat{\tau}^t\right)$

$$
\begin{aligned}
z_a^t &= y_a - \sum_j A_{aj}\theta_j^t \\
&+ \sum_j A_{aj}\left(\frac{\partial \eta^R}{\partial x}\left(\theta_j^{t-1} + \sum_b A_{bj}^* z_b^{t-1}\right)\right) R\left(A_{aj}^* z_a^{t-1}\right) \\
&+ \sum_j A_{aj}\left(\frac{\partial \eta^R}{\partial y}\left(\theta_j^{t-1} + \sum_b A_{bj}^* z_b^{t-1}\right)\right) I\left(A_{aj}^* z_a^{t-1}\right) \\
&+ i\sum_j A_{aj}\left(\frac{\partial \eta^I}{\partial x}\left(\theta_j^{t-1} + \sum_b A_{bj}^* z_b^{t-1}\right)\right) R\left(A_{aj}^* z_a^{t-1}\right) \\
&+ i\sum_j A_{aj}\left(\frac{\partial \eta^I}{\partial y}\left(\theta_j^{t-1} + \sum_b A_{bj}^* z_b^{t-1}\right)\right) I\left(A_{aj}^* z_a^{t-1}\right) \quad \text{for } a = 1, 2, \ldots, n
\end{aligned}
$$

$t = t + 1$

$$
\begin{aligned}
\hat{\tau}^t &= \frac{\hat{\tau}^{t-1}}{\delta} < \Big|\Big| \frac{\partial \eta^R}{\partial x}\left(\mathbf{A}^h \boldsymbol{z}^{t-1} + \hat{\boldsymbol{\theta}}^{t-1}\right) + \frac{\partial \eta^R}{\partial y}\left(\mathbf{A}^h \boldsymbol{z}^{t-1} + \hat{\boldsymbol{\theta}}^{t-1}\right) \\
&+ i\frac{\partial \eta^I}{\partial x}\left(\mathbf{A}^h \boldsymbol{z}^{t-1} + \hat{\boldsymbol{\theta}}^{t-1}\right) + i\frac{\partial \eta^I}{\partial y}\left(\mathbf{A}^h \boldsymbol{z}^{t-1} + \hat{\boldsymbol{\theta}}^{t-1}\right) \Big|\Big| > \\
\hat{\tau}^t &= \sqrt{N_0 + \left(\hat{\tau}^t\right)^2}
\end{aligned}
$$

end while

3.4 Numerical Results

3.4.1 Problem Formalism

In this section, we will present some numerical results for analyzing the performance of OMP, AMP and CAMP algorithms. For the simulation framework in Section 3.4.2, we first consider the "easiest" scenario for OMP, AMP, and CAMP, the noise-free measurement vector with Gaussian sensing matrix $\mathbf{A}$, and then plot the phase transition curve for comparing the performance of all the algorithms.

Next we apply the tap-delayed line channel model (Figure 2.5) to estimate the static channel taps, denoted by $\boldsymbol{h} = [h_1, h_2, \ldots, h_N]^T$. Pilot sequence is denoted by $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]$, then, the received samples $\boldsymbol{y} = [y_1, y_2, \ldots, y_n]^T$ is the convolution between $\boldsymbol{x}$ and $\boldsymbol{h}$, expressed as

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_N & x_{N-1} & x_{N-2} & \ldots & x_1 \\ x_1 & x_N & x_{N-1} & \ldots & x_2 \\ x_2 & x_1 & x_N & \ldots & x_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{N-1} & x_{n-2} & x_{n-3} & \ldots & x_n \end{bmatrix} \cdot \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_N \end{bmatrix} + \boldsymbol{w} \tag{3.41}$$

that is,

$$\boldsymbol{y} = \mathbf{A} \cdot \boldsymbol{h} + \boldsymbol{w} \tag{3.42}$$

where $\boldsymbol{w} = [w_1, w_2, \ldots, w_n]^T$ is the Gaussian noise vector. In this problem setup, the sensing matrix $\mathbf{A}$ is a shifted-orthogonal pilot matrix such that each row of $\mathbf{A}$ is a shifted pilot sequence. To construct a nearly random matrix $\mathbf{A}$, we choose the Zadoff-Chu sequence as pilot signals which has very low autocorrelation [37]. Then phase transition curves are plotted with pilot shifting matrix $\mathbf{A}$ and Rayleigh distributed channel taps in different noise levels.

In the following section 3.4.3, we tested the MSE performance of all the sparse estimation algorithms in OFDM setup with comparison to different bounds for different SNR levels. We follow the problem setup in [19]. The received samples in the frequency domain is

$$\boldsymbol{Y} = \mathbf{H}\boldsymbol{X} + \boldsymbol{W} \tag{3.43}$$

where $\boldsymbol{Y} = [Y_0, Y_1, ..., Y_{N-1}]^T$ is the received OFDM symbols, $\boldsymbol{X} = [X_0, X_1, ..., X_{N-1}]^T$ is the transmitted OFDM symbols, and $\boldsymbol{W} = [W_0, W_1, ..., W_{N-1}]^T$ is the Gaussian

noise. Matrix $\mathbf{H}$ is defined to be a $N \times N$ diagonal matrix containing the components $\text{diag}(\mathbf{H}) = [H_0, H_1, ..., H_{N-1}]$, which is the DFT of $\boldsymbol{h}$ (Severe ICI may cause the channel matrix to be band-diagonal, this effect can also be included). Assume the indexes of pilot symbols are $\boldsymbol{p} = [p_1, p_2, \ldots, p_n]$, then the received symbols corresponding to the pilot sub-carriers' frequencies are

$$\boldsymbol{Y_p} = \mathbf{H}\boldsymbol{X_p} + \boldsymbol{W_p} \tag{3.44}$$

where $\boldsymbol{Y_p} = [Y_{p_1}, Y_{p_2}, ..., Y_{p_n}]^T$, $\boldsymbol{X_p} = [X_{p_1}, X_{p_2}, ..., X_{p_n}]^T$. The pilot density is defined as $\delta = \frac{n}{N}$. To make the channel coefficients $h_l, l = 0, ..., L-1$ appear in time domain, where L is the number of discrete channel taps and in the simulations we set $L = N$, let us define $\mathbf{D}$ to be an $N \times L$ N-point DFT matrix with elements $D_{k,l} = e^{-j2\pi l \frac{k}{N}}$. Hence $H_{k,l} = \sum_{l=0}^{L-1} D_{k,l} h_l$. We also define a matrix $\text{diag}(\mathbf{C_p}) = \boldsymbol{X_p}$ and obtain

$$\mathbf{H}\boldsymbol{X_p} = \mathbf{C_p}\mathbf{D_p}\boldsymbol{h} \tag{3.45}$$

where $\mathbf{D_p}$ is a partial DFT matrix composed of rows in $\mathbf{D}$ with indexes $\boldsymbol{p}$. Then, for processing both pilots and received observations in the frequency domain, the sensing matrix $\mathbf{A}$ can be written as $\mathbf{A} = \mathbf{C_p}\mathbf{D_p}$. At last we can obtain the compressed sensing formula in OFDM setup as

$$\boldsymbol{Y_p} = \mathbf{A}\boldsymbol{h} + \boldsymbol{W_p} \tag{3.46}$$

The pilot sub-carriers $\boldsymbol{X_p}$ is set to be unit one in the simulation, then the sensing matrix $\mathbf{A} = \mathbf{D_p}$ is just a partial DFT matrix. As stated in [19] and analyzed in [38], a uniformly-spaced pilot sub-carriers results to the incoherence of $\mathbf{A}$ (see Equation 3.5) being one and this implies that no sparse signal recovery is possible. For this reason, we set pilot sub-carriers randomly-spaced and it also shows good performance in the simulation in Section 3.4.3.

3.4.2 Phase Transition

Static Real-Valued Channel Estimation

We first obtain the phase transition curve for noise-free measurements with Gaussian sensing matrix $\mathbf{A}$ as a benchmark, then replace $\mathbf{A}$ with pilot-shifting matrix. The simulation parameters are selected as followings:

1. Consider 100 equispaced values of δ and ρ both between 0.1 and 1.

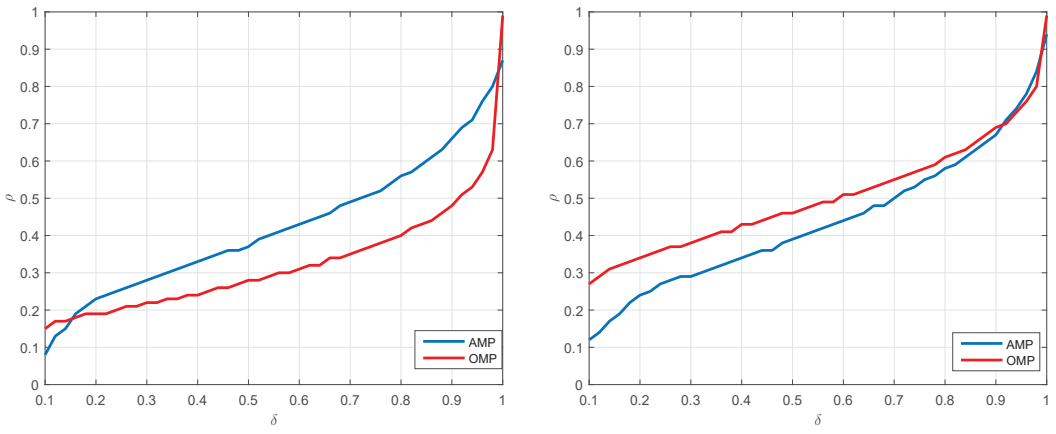
2. For each (δ, ρ) pair, calculate the number of measurements $n = N\delta$, and the number of nonzero channel taps $k = \rho n$, and then make 100 trials.

3. Generate $N = 1000$ discrete channel taps $\boldsymbol{h}$ with k nonzero elements in random position.

4. In each trial, set the MSE tolerance as 10^{-4} and the number of iteration as 1000. A trial is successful if the MSE is lower than the tolerance.

5. For each (δ, ρ) pair, calculate the success rate, and for a fixed value of δ, find the largest ρ that makes the success rate larger than 0.5.

Note that the success rate 0.5 is a standard choice used in references [10, 27], and it simply means that we can possibly recover the sparse vector when the (δ, ρ) pair is above the phase transition curve. We consider two sparse channel vectors for comparison between OMP and AMP with Gaussian sensing matrix, and one is with Rayleigh distributed nonzero entries and another is with nonzero entries in $\{-1, 1\}$. Figure 3.3 shows the phase transition curves for real sparse vector.



(a) Nonzero entries in {-1, 1}

(b) Rayleigh distributed nonzero entries

Figure 3.3: Phase transition curves of AMP and OMP algorithms for real sparse vector with Gaussian sensing matrix. MSE tolerance for a successful trial is 10^{-4}.

Each phase transition curve partitioned the $\delta - \rho$ plane into two regions. The region below the curve is the successful region where the algorithms can recovery

the signal with a specific tolerance of MSE, and the region above the curve is where the algorithms fail. For example, in Figure 3.3(a), when $\delta = 0.6$, the sparsity level that AMP can tolerance is about $\rho = 0.4$, which means that the algorithm can recover the signal with $\rho\delta = 0.24$ percentage nonzero elements with success probability larger than 0.5. Above this percentage, AMP will fail with probability larger than 0.5. Phase transition gives us a good guidance for setting the under-sampling rate in a communication system. It is important to note that AMP has the same phase transition for those two distribution of nonzero elements while OMP has a bad performance when nonzero entries in $\{-1, 1\}$. Actually, the performance of AMP has been rigorously proved that it is not influenced by the distribution of sparse vectors [33]. For OMP, it suffers from recovering the nonzero entries with equivalent power because at each iteration, there may be two columns of $\mathbf{A}$ that can be chosen since the products between residual and those two columns of $\mathbf{A}$ could be the same. Therefore, OMP has the possibility to pick a wrong column at some iteration.

Gaussian sensing matrix is the ideal case for AMP and OMP, however, it cannot be realized in practice. Therefore, we replace the Gaussian matrix $\mathbf{A}$ with pilot-shifting matrix, and set the MSE tolerance to 10^{-3}. Results are shown in Figure 3.4. Here, we only implemented Rayleigh distributed nonzero channel taps since it is close to the reality.

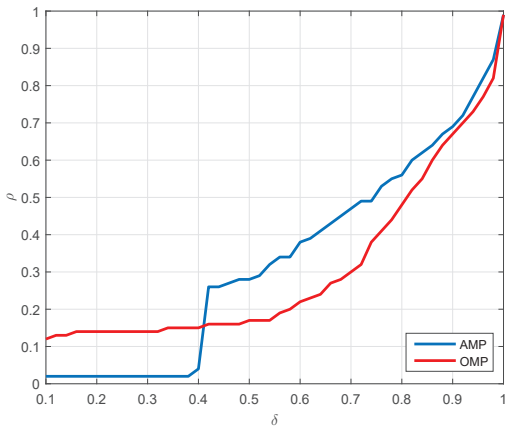


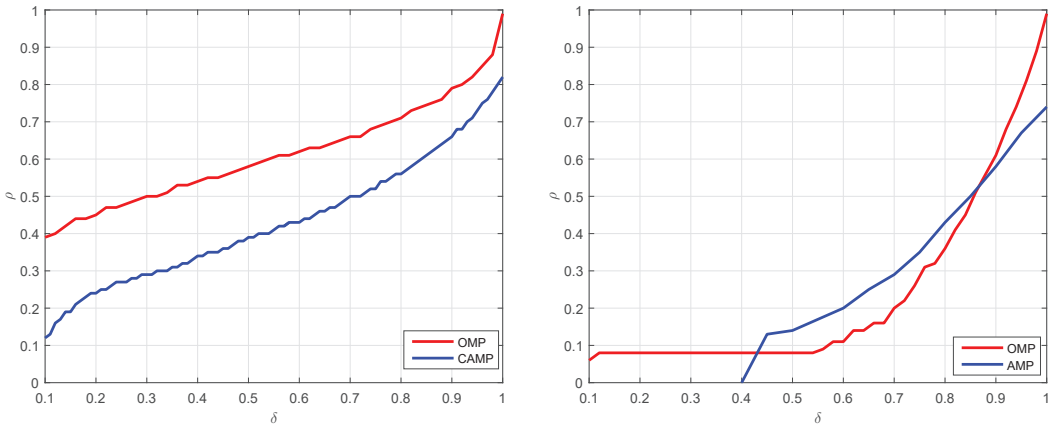
Figure 3.4: Phase transition curves of AMP and OMP algorithms for real sparse vector with pilot-shifting sensing matrix. MSE tolerance for a successful trial is 10^{-3}.

We find that even the pilot sequence has very low autocorrelation, however, it does

not suffice the low correlations between the columns of sensing matrix $\mathbf{A}$, and this leads to the worse performance of AMP and OMP than implementing the Gaussian matrix, especially for the small δ. Note that for $\delta < 0.4$, the sparsity level for AMP that can reach is nearly zero, which means at most of the time AMP cannot work in this case.

Static Complex-Valued Channel Estimation

For the complex case, we follow the same simulation steps and parameter settings as for the real case. Both channel taps and pilot sequence are complex numbers. Figure 3.5 presents the phase transition curve for AMP and OMP algorithm in complex number setting with Gaussian sensing matrix and pilot-shifting sensing matrix, respectively .



(a) Gaussian sensing matrix (b) Pilot-shifting sensing matrix

Figure 3.5: Phase transition curves of CAMP and OMP algorithms for complex sparse vector with Gaussian sensing matrix. The nonzero elements are Rayleigh distributed, and MSE tolerance for a successful trial is 10^{-4} for Gaussian sensing matrix and 10^{-3} for pilot-shifting matrix.

It is same for complex numbers that when the sensing matrix is the pilot-shifting matrix, both CAMP and OMP have bad performances for small under-sampling values. To partially compensate for this, for CAMP, we add an artificial random part to the sensing matrix $\mathbf{A}$ in the estimation procedure in the CAMP algorithm. This artificial random part is a Gaussian matrix with variance $a^2 = 10^{-8}$, and we find that in the simulation it can improve the convergence of CAMP algorithm. However,

to reach the performance of the Gaussian matrix, a better orthogonal pilot sequence with low autocorrelation is needed.

In the OFDM setup (see Section 3.4.1), the sensing matrix is a partial DFT matrix and we simulated phase transition curves in this scenario. The results are in Figure 3.8.

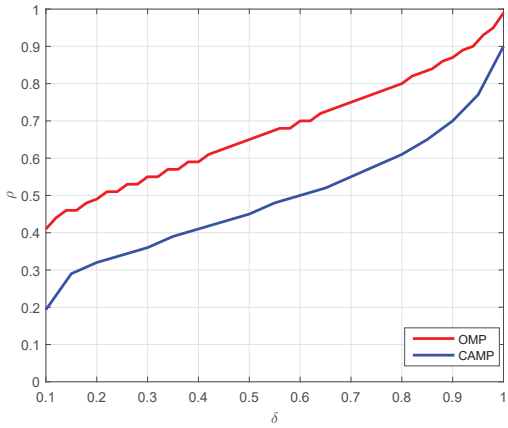


Figure 3.6: Phase transition curves of CAMP and OMP algorithms for complex sparse vector with partial DFT sensing matrix in OFDM setup. The nonzero elements are Rayleigh distributed, and MSE tolerance for a successful trial is 10^{-4}.

Comparing Figure 3.8 and Figure 3.5(a), we can notice that partial DFT sensing matrix has almost the same result as Gaussian sensing matrix, and in both cases, OMP is a better choice for channel estimation. In the following section, we will present more detail about comparison between CAMP and OMP algorithm in OFDM setup.

3.4.3 MSE Performance of Static Channel Estimation in OFDM Systems

To obtain the MSE versus SNR curve in the OFDM setup, we implemented the parameter setting as following:

1. Generate the exactly sparse Rayleigh distributed channel with $L = N = 100$ taps and the number of nonzero taps is $s = 12$.

2. Construct the OFDM sensing matrix $\mathbf{A} = \mathbf{C}_{\boldsymbol{p}}\mathbf{D}_{\boldsymbol{p}}$ with unit one pilot sub-carriers and randomly position vector $\boldsymbol{p}$. The number of pilot sub-carriers is $N_p = \delta N$, where δ is the pilot density. Matrix $\mathbf{A}$ is of size $N_p \times N$.

3. Choose the candidate pilot densities $\delta = [30\%, 40\%, 50\%, 60\%, 70\%, 80\%]$ for simulations. For each δ, we run all the estimation algorithms for 10^4 trials for each SNR, and then calculate the MSE averaging over all the trials and nonzero channel taps.

We choose the OMP algorithm with RDR stopping criteria, denoted by "OMP-RDR". Other different bounds are computed as following based on the technical report [39]:

1. Least Squares (LS): It minimizes the squared measurement error and its solution is

$$\begin{aligned} \hat{\boldsymbol{h}}_{\mathrm{LS}} &= \arg\min_{\boldsymbol{h}} ||\boldsymbol{Y_p} - \mathbf{A}\boldsymbol{h}||^2 \\ &= (\mathbf{A}^h\mathbf{A})^{-1}\mathbf{A}\boldsymbol{Y_p} \end{aligned} \tag{3.47}$$

2. Oracle Estimator: This estimator assume that it knows the priori information about the position of nonzero channel taps, denoted by $\boldsymbol{l} = [l_1, l_2, \ldots, l_s]$. Following the LS estimator but applying the sub-matrix of $\mathbf{A}$ with column indexes $\boldsymbol{l}$, its formula is

$$\hat{\boldsymbol{h}}_{\mathrm{ORL}} = (\mathbf{A}_{\boldsymbol{l}}^h\mathbf{A}_{\boldsymbol{l}})^{-1}\mathbf{A}_{\boldsymbol{l}}\boldsymbol{Y_p} \tag{3.48}$$

3. The Cramer-Rao Lower Bound (CRLB): The CRLB indicates that the variance of the minimum variance unbiased (MVU) estimator is not smaller than the inverse of the Fisher Information. For a linear model as presented in Equation 3.46 but with matrix $\mathbf{A}_{\boldsymbol{l}}$, the CRLB is derived in [40, Section 4.3],

$$\mathbf{C}_{\hat{\boldsymbol{h}}} \geqslant N_0(\mathbf{A}_{\boldsymbol{l}}^h\mathbf{A}_{\boldsymbol{l}})^{-1} \tag{3.49}$$

Then the MSE lower bound for nonzero channel taps is the mean of the diagonal elements in $\mathbf{C}_{\hat{\boldsymbol{h}}}$ in columns $\boldsymbol{l}$, expressed as

$$\mathrm{MSE}_{\mathrm{CRLB}} = \frac{\sum \mathrm{diag}(\mathbf{C}_{\hat{\boldsymbol{h}}})}{N_p} \tag{3.50}$$

4. Matched Filter (MF) Estimator for Exactly Determined System: In the exactly determined system, the sensing matrix $\mathbf{A}$ is compose of 100% pilot sub-carriers,

therefore, $\mathbf{A}$ becomes a DFT matrix $\mathbf{D}$ and is orthonormal, i.e. $\mathbf{D}^h\mathbf{D} = \mathbf{I}$. Then the MF estimator is

$$\hat{\boldsymbol{h}}_{\mathrm{MF}} = \mathbf{D}^h \boldsymbol{Y}_{\mathrm{MF}} \tag{3.51}$$

where $\boldsymbol{Y}_{\mathrm{MF}} = \mathbf{D}\boldsymbol{h} + \boldsymbol{W}$. The LS estimator for the exactly determined system is the same because of the orthonormality of matrix $\mathbf{D}$.

5. The CRLB for Exactly Determined System: It has the same formula as Equation 3.49 with replacing the matrix $\mathbf{A}$ with the orthonormal DFT matrix.

The curves of OMP, LS and lower bounds are formulated in the report [39], and we formulated CAMP for comparison. Figure 4.6(a) shows that OMP and CAMP has a relatively large error floor due to the insufficient pilot density 30%, and the gap between the lower bounds for the underdetermined and exactly determined system is around 6dB. When the pilot density reaches 50%, the performance of OMP and CAMP is almost near the lower bounds for under-determined system, and for pilot density greater than 50%, the MSE improvement of OMP and CAMP is less than 2dB. This implies that we can reach almost the same MSE performance with much less pilot densities, and larger data rate.

For comparison between OMP and CAMP, Figure 4.6 exhibits that OMP is reasonable choice for fast, powerful, and simple sparse estimation algorithm to implement. CAMP can also be implemented for severe noise scenarios.

3.4.4 The Impact of Filtering and Resolution on Static Channel Estimation in OFDM Systems

The results in this section is a part of work in [19]. In this paper, we have studied the influence of roll-off factor β of transmitter-receiver filter (see Figure 2.4) and data rate $\frac{1}{T}$ on the BER performance of sparse channel estimation algorithms, where T is OFDM sample duration. The main result is that a high filtering bandwidth is suggested to preserve the sparsity of the analog channel impulse response and then benefit sparse estimation algorithms. This result is motivated by the super-resolution theory in [41] that the necessary and sufficient condition for unique recovery of the analog object with l_1 minimization is:

$$\tau_{\min} \geqslant \frac{4}{B} \tag{3.52}$$

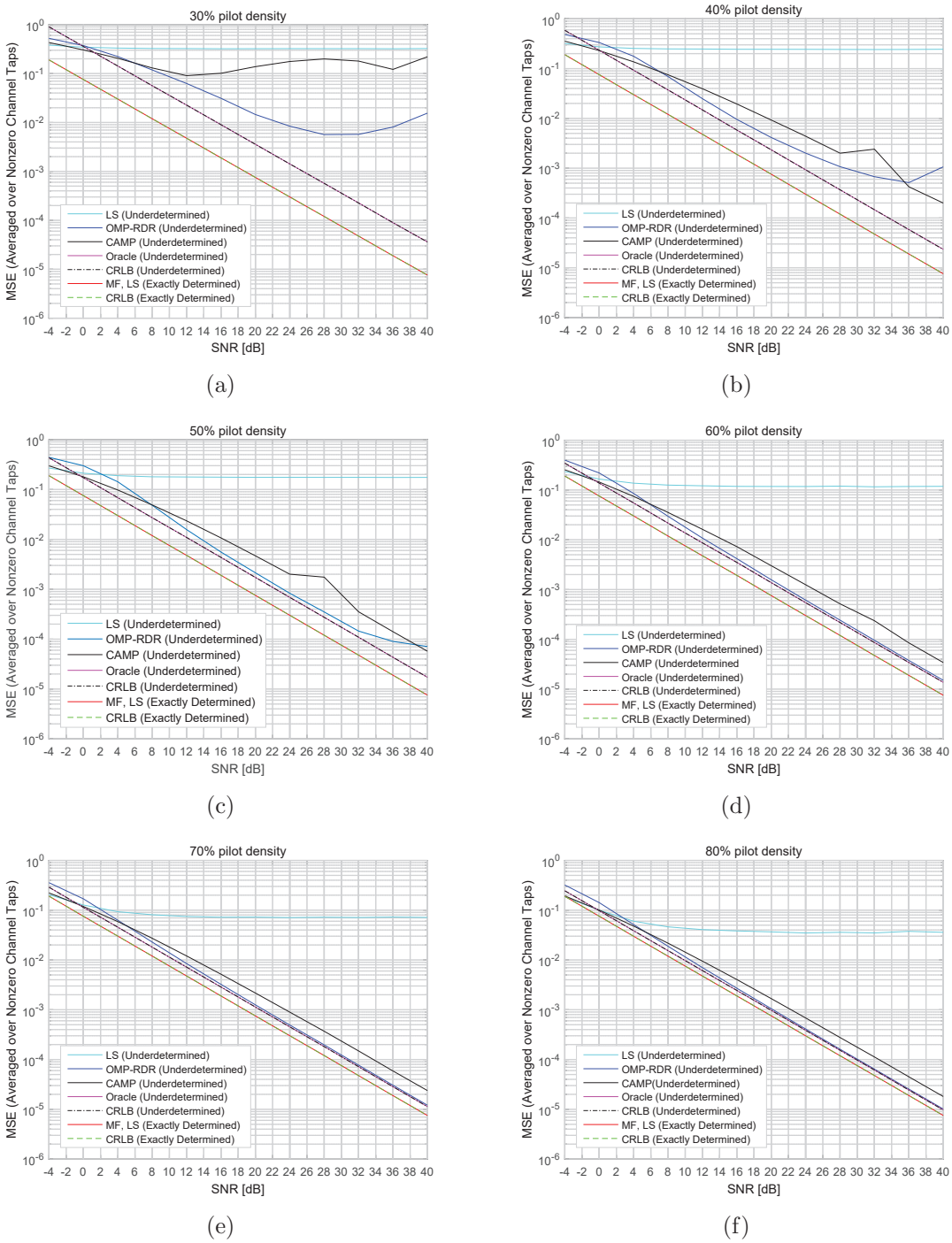


Figure 3.7: MSE performance of OMP and CAMP in OFDM setup with comparison to different lower bounds. MSE curves are plotted for pilot density $\delta = [30\%, 40\%, 50\%, 60\%, 70\%, 80\%]$ respectively. The original channel taps are Rayleigh-distributed and have 100 taps with 12 nonzero elements. MSE is averaged over all the non-zero channel taps.

where $\tau_{\min}$ is the relative minimum delay and $B = \frac{1+\beta}{T}$ is the system bandwidth. Through extensive experiments, we have found that a compact transmitter-receiver filter is crucial to the performance of channel estimation and $\beta = 1$ is a reasonable option since the filters have the smallest side-lobes in the time domain in this case.

We interpreted the results in [19] (see figure 3 in this paper) in a perspective of phase transition, and explored what combination of β and $\frac{1}{T}$ gives us the most sparse channel. The result is in Figure 3.8, and experiment details can be found in [19].

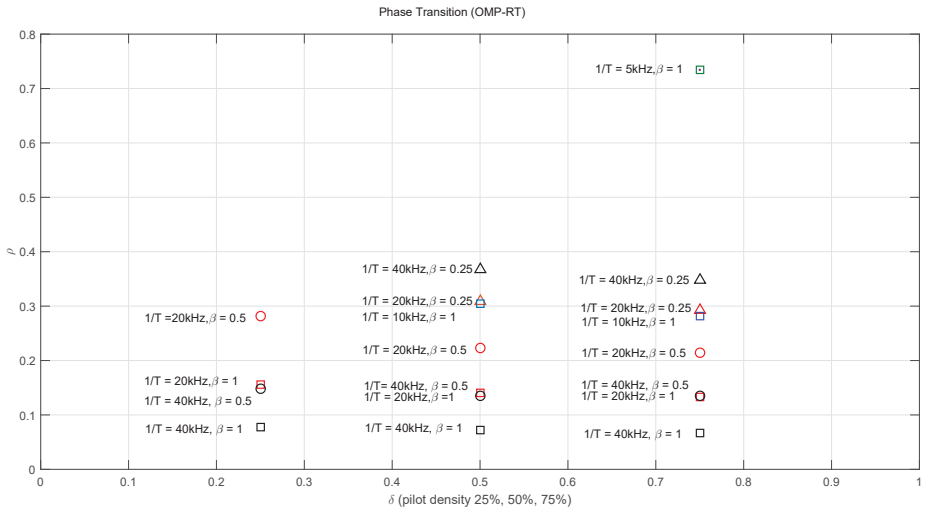


Figure 3.8: Phase transition curve of OMP-RT algorithm for static channel estimation in OFDM system. The BER thresholding is 10^{-3} and $\tau_{\min} = 0.1$ms. We have tested three values of $\delta = [0.25, 0.5, 0.75]$ and the values of ρ are found for each combination of β and $\frac{1}{T}$ by computing the number of channel taps whose amplitudes are above the noise variance.

We have run OMP-RT algorithm for sparse channel estimation in OFDM system with pilot density $\delta = [0.25, 0.5, 0.75]$ and obtain its BER performances for each value of δ. With setting BER thresholding as 10^{-3}, we tested the sparsity level of discrete-time channel taps for 12 combinations of $\beta = [0, 0.5, 1]$ and $\frac{1}{T} = [5\text{kHz}, 10\text{kHz}, 20\text{kHz}, 40\text{kHz}]$.

For pilot density $\delta = 0.25$, there are only 4 combinations of β and $\frac{1}{T}$ that can reach BER $= 10^{-3}$. Among those 4 cases, the discrete channel is the most sparse when $\frac{1}{T} = 40$kHz and $\beta = 1$. For other δ values, we have the same results. As δ goes higher, we can see that OMP-RT is successful for more cases since the algorithm can tolerate more dominant channel taps with more measurements. It is important to note

that for each pair of δ and $\frac{1}{T}$, the case where $\beta = 1$ always give us the channel with the highest sparsity level and the "easiest" channel for sparse estimation algorithms. We cannot even find a point with $\beta = 0$ in this figure because in this scenario, the number of dominant channel taps is too much to make OMP-RT algorithm success with BER thresholding equal to 10^{-3}. On the other hand, we can also find that the higher date $\frac{1}{T}$ we have, the more sparse channel we get.

Hence, we conclude that a transmitter-receiver filter with $\beta = 1$ can give us a relatively more sparse channel to benefit sparse channel estimation algorithms, and higher date rate is required with combination of β to satisfy the super-resolution theory.

Chapter 4

Sparse Channel Tracking

4.1 Introduction

In contrast to the simulations in Chapter 3, the time-varying UWA channel estimation will be considered. The speed of the channel variation can be parametrized by the spreading factor $\frac{\tau_{\max}}{T_{\text{coh}}}$ (see Section 2.3), and if it is larger than 1, then it means that the channels are over-spread and vary rapidly. In a fast varying channel environment, only a single carrier communication system is feasible to implement since the Doppler spread would lead to severe ICI in a multi-carrier system. We explore the window-based tracking method in this scenario, which is based on reference [5] but we apply the RLS adaptive filter as a performance benchmark.

For communicating via a mildly varying channel with a multi-carrier system, the Doppler effect still needs to be considered. For compensating the Doppler shift due to the motion of transmitters and receivers, a popular method is to first resample the received signals and then compensate for the estimated residual Doppler shift [42]. On the other hand, if the transmission system is OFDM, after the Doppler shift compensation, the ICI caused by the Doppler spread will result to a mixed channel frequency response matrix $\mathbf{H}$ that is no longer diagonal as expressed in Equation 3.43, and further, it will lead to higher errors in the symbol detection process. To solve this problem, some authors considered the banded diagonal structure of the mixed channel matrix $\mathbf{H}$ and introduced more complex receiving process and equalization algorithms [43, 44, 26, 45]. Experiments were also implemented in those papers to test how many banded diagonal elements of $\mathbf{H}$ to need be considered sufficient.

Different from the approaches mentioned above, we assume that in an OFDM system the channel taps keep constant during each OFDM symbol. Then we apply the Kalman filter to sequentially track the channel symbol by symbol with a dynamic state model being constructed in the first two OFDM symbols. At the receiver part, the channel frequency response matrix $\mathbf{H}$ is still diagonal. We simulated the BER

performance for different pilot densities at different time-varying levels of the channel which is measured by the shape parameter α of the exponential form Doppler spectra (see Section 2.1.3), and we analyzed how much channel variation is tolerable for OFDM system. The Kalman filter applied is sparse-aided, that is, it only tracks the channel taps that have significant power. This technique has been used in [11], but we applied a different problem setup whose detail is in 4.3. The results of the channel estimations in the first two OFDM head symbols will obtain the information about the position of the significant channel taps.

In the next Section 4.2, the window-based tracking method will be introduced. The main idea is that we assume the channel taps slowly vary during a short time interval and estimate the channel in each time interval. OMP, LS, and RLS algorithms will be implemented.

The following Section 4.3 will describe the channel tracking mechanism in an OFDM system, and Kalman filter will be introduced. OMP, LS, and RLS will also be compared for the BER performance.

4.2 Channel Tracking in Single Carrier Communication System

4.2.1 Window-Based Channel Tracking

We apply the basic channel model stated in Section 2.1.3 with each analog path generated by the SOS model and each channel tap in the discrete domain is obtained after the transmitter and receiver filtering, denoted by $\boldsymbol{h} = [h_1, h_2, \ldots, h_N]^T$. Consider the tapped-delay line channel model, and the time-varying channel taps are denoted by $h_l[t]$ which represents the value of the lth channel tap at the sampling time t. Both subscripts are integers. In the discrete time domain, the received signals $\boldsymbol{y} = [y_1, y_2, \cdots, y_N]^T$ can be expressed as

$$
\begin{aligned}
&y_1 = h_1[N]x_N + h_2[N-1]x_{N-1} + h_3[N-2]x_{N-2} \cdots + h_N[1]x_1 + w_1 \\
&y_2 = h_1[N+1]x_1 + h_2[N]x_N + h_3[N-1]x_{N-1} + \cdots + h_N[2]x_2 + w_2 \\
&\vdots \\
&y_n = h_1[N+n]x_{n-1} + h_2[N+n-1]x_{n-2} + h_3[N+n-2]x_{n-3} + \cdots + h_N[n]x_n + w_n \\
&\vdots
\end{aligned}
\tag{4.1}
$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_N]$ is the Zadoff-chu pilot sequence and $\boldsymbol{w} = [w_1, w_2, \ldots, w_N]$ is the Gaussian noise. We can compare the Equation 4.1 with Equation 3.41 and it is easy to see that the received signals are no longer the convolution between the channel taps and pilot sequences due to the changing of the channel taps. To estimate the channel taps $\boldsymbol{h}$ at each time sample with knowing $\boldsymbol{x}$ and $\boldsymbol{y}$, we stack W successively received samples or equivalently W successive equations. Here W is the size of the observation window. The window is slid by one sample each time and the channel estimation algorithms are conducted within each window. For example, to estimate the channel taps at first sampling time, we stack the first W equations, and assume that the channel taps within the first window remain constant. That is,

$$\begin{aligned}
&\big\{h_1[N], h_1[N+1], ..., h_1[N+W]\big\} \approx \tilde{h}_1 \\
&\big\{h_2[N-1], h_2[N], ..., h_2[N+W-1]\big\} \approx \tilde{h}_2 \\
&\vdots \\
&\big\{h_N[1], h_N[2], ..., h_N[W]\big\} \approx \tilde{h}_N
\end{aligned} \tag{4.2}$$

and we use $\tilde{h}_1, \tilde{h}_2, \cdots, \tilde{h}_N$ to represent the approximate contant channel taps in the first window. It is obvious that the smaller the size of the window we choose, the more accurate the estimation results.

Now, taking an example of the estimation in the first window, stacking the first W equations of 4.1, we rewrite them in matrix formula as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_W \end{bmatrix} = \begin{bmatrix} x_N & x_{N-1} & x_{N-2} & \ldots & x_1 \\ x_1 & x_N & x_{N-1} & \ldots & x_2 \\ x_2 & x_1 & x_N & \ldots & x_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{N-1} & x_{W-2} & x_{W-3} & \ldots & x_W \end{bmatrix} \cdot \begin{bmatrix} \tilde{h}_1 \\ \tilde{h}_2 \\ \tilde{h}_3 \\ \vdots \\ \tilde{h}_N \end{bmatrix} + \boldsymbol{w} + \boldsymbol{v} \tag{4.3}$$

That is,

$$\boldsymbol{y}_{1,W} = \mathbf{A}_{1,W} \cdot \tilde{\boldsymbol{h}} + \boldsymbol{w} + \boldsymbol{v} \tag{4.4}$$

Here $\boldsymbol{y}_{1,W}$ represents the first window measurement vector of size $W \times 1$, $\mathbf{A}_{1,W}$ is the $W \times N$ measurement matrix in the first window, $\tilde{\boldsymbol{h}}$ is a $N \times 1$ vector that need to be estimated and $\boldsymbol{v}$ is the noise due to the varying channel taps averaging within the window. Similarly, the measurement vector in the second window would be

$\boldsymbol{y}_{2,W+1} = [y_2, y_3, \ldots, y_{W+1}]^T$, and in the third window $\boldsymbol{y}_{3,W+2} = [y_3, y_4, \ldots, y_{W+2}]^T$, and so on.

Now the compressed sensing formula is constructed in Equation 4.4, and the sparse estimation algorithms (OMP, AMP and CAMP) can be implemented for each window. Let $\delta = \frac{W}{N}$ and $\rho = \frac{k}{W}$ denote the undersampling parameter and the sparsity parameter, where k is the approximate number of non-zero elements in $\tilde{\boldsymbol{h}}$. The estimation within each window is actually a trade-off between the window size and the success rate of sparse estimation algorithms. For better performance of those algorithms, we expect large window size, however, it could lead to less accurate assumption of $\tilde{\boldsymbol{h}}$. For rapid time-varying channel, the window size will be much smaller, as a consequence, the sparse estimation algorithms could fail because of insufficient data to operate. Therefore, we apply the RLS adaptive filter to capture the time-varying behavior of the channel.

4.2.2 Recursive Least Squares (RLS) Algorithm

RLS algorithm is also known as the sequential least squares. Different from the classical least squares (see Equation 3.47) which obtain the solution of $\boldsymbol{h}$ using all the measurements $\boldsymbol{y}$, it updates the estimate of $\boldsymbol{h}$ with the new incoming received sample. This procedure avoids solving the linear equation 3.47 at each sampling time and therefore it decreased the computation complexity with avoiding the matrix inversion.

The motivation of RLS is to solve the exponential weighted least squares problem

$$\hat{\boldsymbol{h}}_{\mathrm{LS}} = \arg\min_{\tilde{\boldsymbol{h}}} \sum_{i}^{n} \lambda^{n-i} \left(\boldsymbol{y}[i] - \hat{\boldsymbol{y}}[i]\right)^2 \tag{4.5}$$

where $\hat{\boldsymbol{y}} = \mathbf{A} \cdot \hat{\boldsymbol{h}}_{\mathrm{LS}}$ is the estimated received samples within each window. We have omitted the subscript window index for convenience. We need to find the solution that minimizes the weighted measurement errors with applying the weighting factor or forgetting factor $\lambda (0 < \lambda < 1)$. In this way we can put more weights on the new incoming data and reduce the influence of old data, and the smaller λ we have, the faster the algorithm "forgets" the old data.

The RLS algorithm iterates through two main steps, estimator update and covariance update. Following the derivation in reference [46], they are summarized in

Algorithm 4.

Algorithm 4 Recursive Least Squares (RLS)

Input:

A scaler $\boldsymbol{y}[n]$ as the current received sample

A scaler λ as the forgetting factor

A $N \times 1$ vetor $\boldsymbol{x}[n]$ as the current transmitted pilot signals

A $N \times 1$ vector $\boldsymbol{h}[-1]$ as the initialization of estimate

A $N \times N$ diagonal matrix $\mathbf{P}[-1] = \alpha\mathbf{I}$ as the initialization of covariance matrix of the estimate

Output:

An estimate $1 \times N$ vector $\hat{\boldsymbol{h}}$ at each sampling time

1: Initialization: $n = 0$

2: **while** $n \leq$ Total number of samples **do**

3: $\boldsymbol{K}[n] = \frac{\mathbf{P}[n-1]\boldsymbol{x}^*[n]}{\lambda + \boldsymbol{x}^T[n]\mathbf{P}[n-1]\boldsymbol{x}^*[n]}$ (Gain calculation)

4: $\hat{\boldsymbol{h}}[n] = \hat{\boldsymbol{h}}[n-1] + \boldsymbol{K}[n]\left(\boldsymbol{y}[n] - \boldsymbol{x}^T[n]\hat{\boldsymbol{h}}[n-1]\right)$ (Estimator update)

5: $\mathbf{P}[n] = \left(\mathbf{I} - \boldsymbol{K}[n]\boldsymbol{x}^T[n]\right)\lambda^{-1}\mathbf{P}[n-1]$ (Covariance update)

6: $n = n + 1$

7: **end while**

Here, the index n represents the current discrete sampling time. The estimator update is the summation of the previous estimate and a correction term. The correction term is computed with the current received sample and a gain vector which represents our confidence in the new data. If the forgetting factor λ is large, then the gain vector will be small and therefore the estimator will be updated with less correction. The covariance matrix $\mathbf{P}[n-1]$ is also the factor that can influence the gain vector. If $\mathbf{P}[n-1]$ is large, which means we do not have much confidence on the previous estimate, then the gain vector will be large and as a consequence, the estimator update step will apply larger correction term.

In the simulation, we initialize $\hat{\boldsymbol{h}}[-1]$ with the OMP estimation results in the first 10 windows. Hence, the value of α in the initialization of covariance matrix $\mathbf{P}[-1]$ can be chosen with a relatively small value ($\sim 10^{-2}$) since we did not start RLS with all-zero vector and we have some confidence on $\hat{\boldsymbol{h}}[-1]$. However, to minimize the biased effect we could still choose large α, the RLS will work as the same.

4.2.3 Numerical Results

We compare the MSE performance of each algorithm averaging over time and the simulation steps are as following:

1. Generate the time-varying channel taps as stated in Section 2.1.3. The shape parameter α is to control the channel varying speed. All the other parameter settings are the same as the example in Section 2.1.3. The maximum delay length (the number of channel taps to estimate) is denoted by $L_{\max}$.

2. Specify the window size $W = \delta L_{\max}$.

3. For initializing RLS algorithms, we apply the OMP estimate results as $\hat{\boldsymbol{h}}[-1]$.

4. For each SNR value, run the algorithms in each window and then calculate the averaged MSE for only nonzero channel taps. The threshold to determine the nonzero taps is 10^{-4}.

We implemented two methods. One is to track all the channel taps, and the other is to only track the dominant channel taps and we call this method as "sparse-aided" tracking. For the sparse-aided tracking, in the first 10 windows, we use OMP to estimate the positions of the dominant channel taps with window size same as the number of the channel taps. Then, we decreased the window size and the number of rows of the pilot-shifting sensing matrix also shrank. Let us denote the position of dominant taps as $l_1, l_2, ..., l_s$ whose amplitudes are above 10^{-4}, and we find the smallest window size N_1 which satisfies $mN_1 \neq l_i - l_j$, where $i, j \in (1, 2, ..., s)$ and m is a integer. Then, we sent the pilot sequence with period N_1, and therefore the pilot-shifting matrix $\mathbf{A}$ is of size $N_1 \times N$. In this way, the columns of matrix $\mathbf{A}$ with indexes $l_1, l_2, ..., l_s$ have zero autocorrelation. In simulations, we also tested some window sizes smaller than N_1 and we will see that the window size will not influence the results of RLS algorithm but have a significant impact on OMP.

Fig. 4.1 shows the tracking behavior of the OMP and RLS algorithms of an instantaneous run when SNR = 40 dB, $\alpha = 5$, and $\delta = 1$. This is the all-tap tracking method and we choose three candidate channel taps to compare the results. In Fig. 4.1(a), we plot the time evolution of the first channel tap that has large amplitude relative to other ones and we can see that both algorithms can track it very well. Fig. 4.1(b)

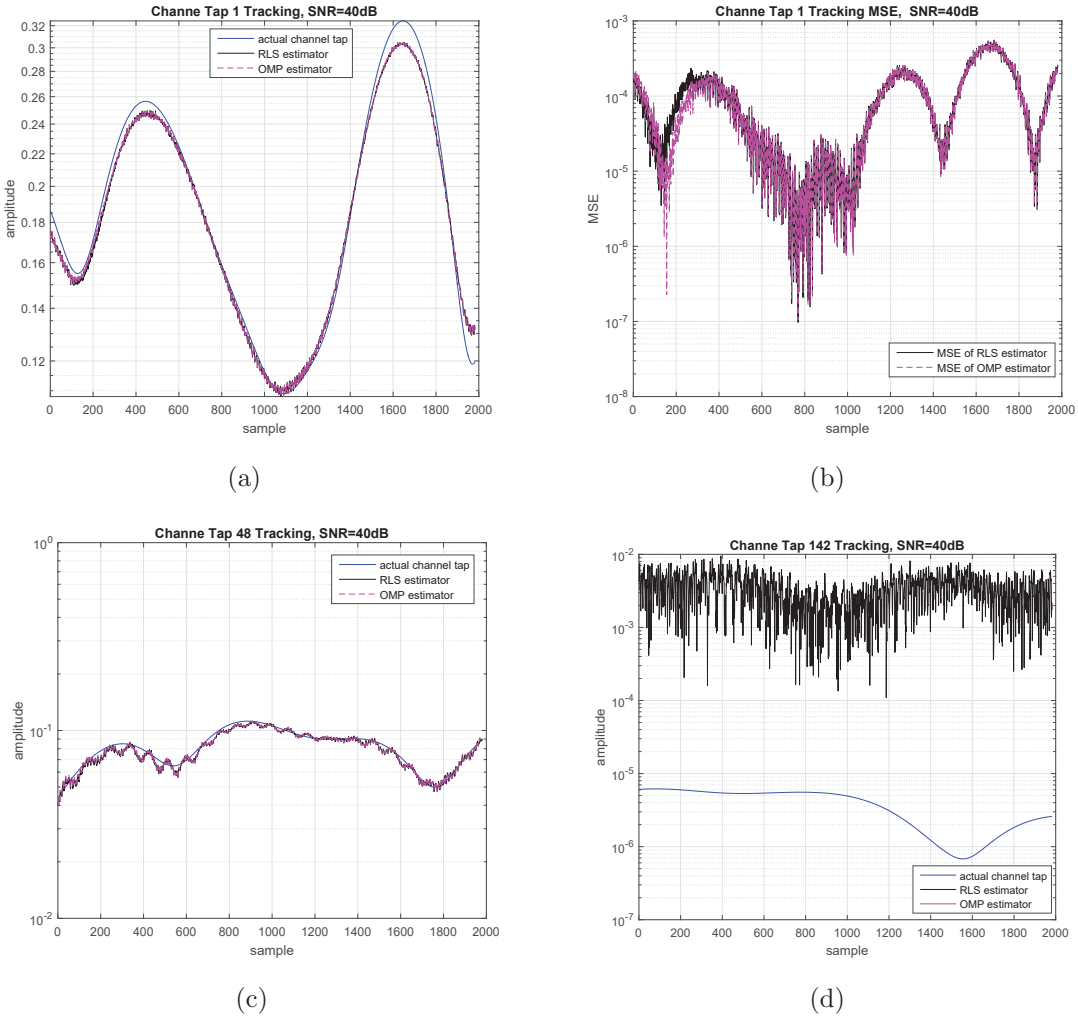
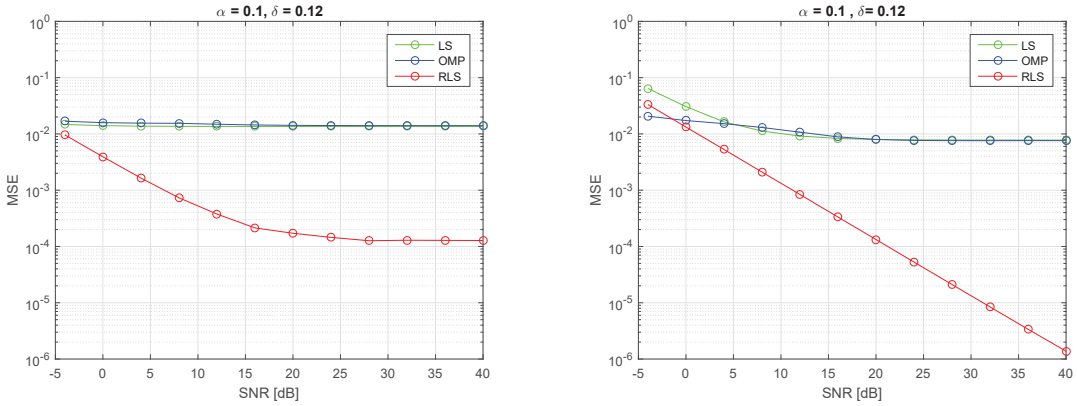


Figure 4.1: Tracking behaviors of RLS and OMP algorithms on time-varying channel in single-carrier communication system with SNR = 40 dB, $\alpha = 5$ and $\delta = 1$. The number of discrete channel taps is 260 and the number of discrete time samples is 2000. The x-axis in all the subfigures is the sample index. Figure (a) shows the time evolution of the original first channel tap and RLS, OMP estimated results, and figure (b) is the corresponding MSE curve. Figure (c) and (d) are the 48th and 142th channel tap tracking performance respectively.

shows the tracking MSE over the sampling time. Fig. 4.1(c) presents the time evolution of a relatively small channel tap, and we can see that both algorithms are still able to track it. However, in Fig. 4.1(d) we focus on a channel tap which is nearly equal to zero, and we find that RLS algorithm fails to estimate the amplitude and OMP estimates it as a zero channel tap, which is why we cannot see the OMP curve in the figure.

To test the tracking performance of RLS, OMP and LS (as a benchmark) algorithms with different noise levels, we plot the MSE curve versus SNR with different values of α. We compute the average MSE over all the dominant channel taps whose amplitudes are above 10^{-4}. If we computed MSE over all the taps, we will have smaller MSE results. We first compare the results between all-tap tracking method and sparse-aided tracking methods (see Figure 4.2). For the sparse-aided tracking method, the size of matrix $\mathbf{A}$ shrinks and it gives benefits to all the algorithms. However, when $\delta = 0.12$ and the number of channel taps is $L_{\max} = 260$, then the window size is $W = \delta L_{\max} \approx 31$, therefore, comparing with the number of dominant channel taps being 50 approximately in the simulation, the compressed sensing problem is still an under-determined problem. As a consequence, we can see that in Figure 4.2 there is a large gap between the performance of RLS and the other two estimation algorithms (OMP and LS).



(a) All-tap tracking

(b) Sparse-aided tracking

Figure 4.2: Comparison of MSE performance between all-tap tracking method and sparse-aided tracking method when $\alpha = 0.1$ and $\delta = 0.12$. Here, the number of channel taps is $L_{\max} = 260$ and window size is calculated by $W = \delta L_{\max} \approx 31$. In both subfigures, MSE curves are plotted for RLS, OMP and LS algorithms over SNR value between -4dB and 40dB.

In order to test the performance of sparse-aided tracking method further, we plotted MSE curves with different window sizes ($\delta \in [0.12, 0.2, 0.39]$) and different values of α ($\alpha \in [0.1, 1, 5]$) in Figure 4.3. Figure 4.2 (b) is also included in the subfigures for convenience.

We can notice that RLS algorithm barely influenced by the size of tracking window

while OMP and LS suffer from the under-determinant formalism when $\delta \in [0.12, 0.2]$. Furthermore, from the phase transition curve in Section 3.4.2, we can see that OMP has small probability to recover the channel tap when δ is small and sensing matrix $\mathbf{A}$ is a pilot shifting matrix. This is also another important reason why OMP has a bad performance for tracking the channel when the window size is much smaller than the number of the discrete channel taps. For any α value, RLS has the best MSE performance, however, it still has some penalty when $\alpha > 1$ corresponding a fast varying channel. This is because even when $\delta = 0.12$ and the window size is small, the channel tap variation inside each window cannot be ignored.

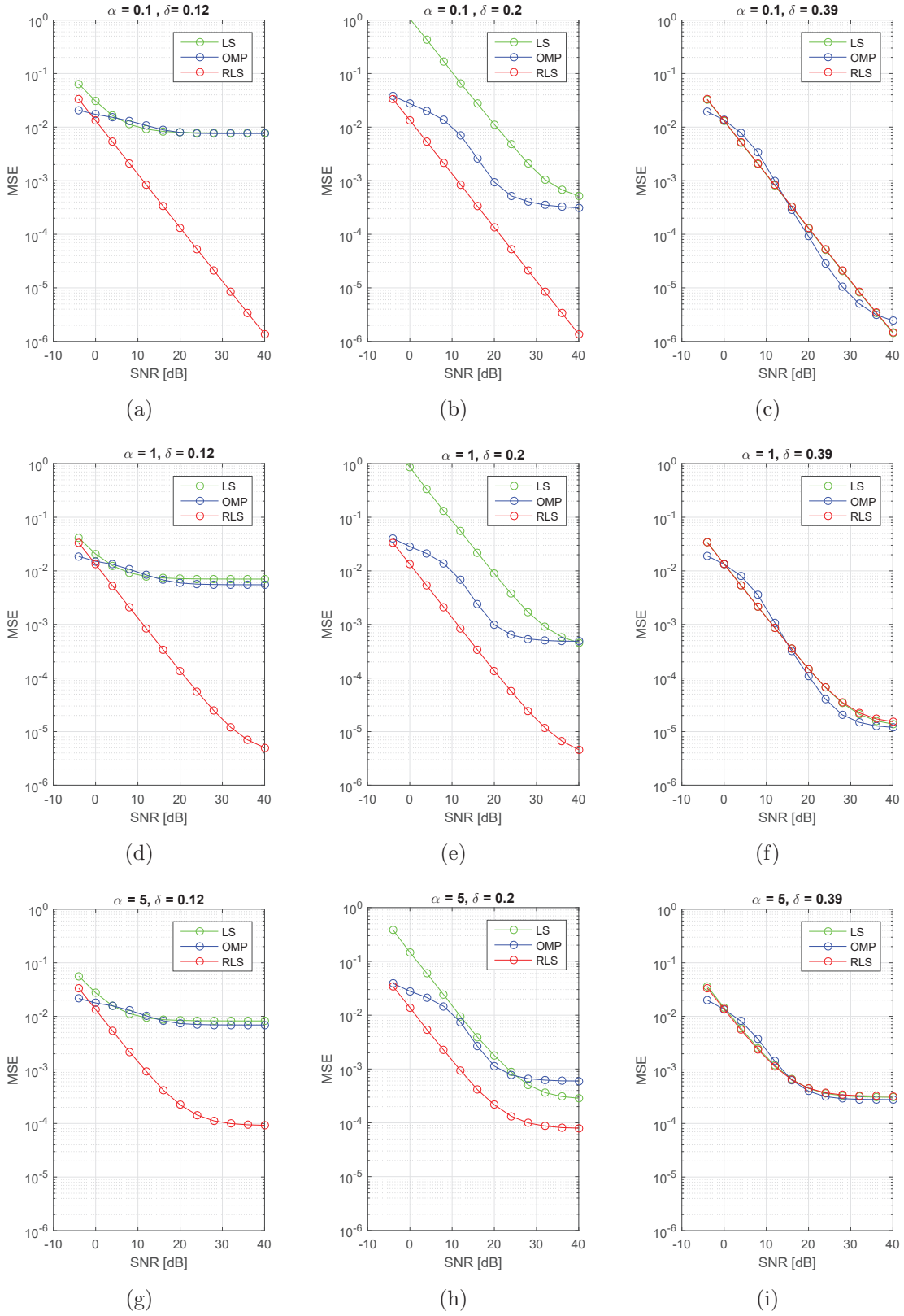


Figure 4.3: Comparison of MSE performances of algorithms RLS, OMP and LS for time-varying channel tracking. MSE is computed over all dominant channel taps with applying spares-aided tracking method.

4.3 Channel Tracking in OFDM Communication System

4.3.1 Tracking Mechanism

Different from the previous section where the channel taps are updated for each received sample, we consider about the estimation updated for each OFDM symbol duration in this section. Assume that there are L channel taps and the number of OFDM subcarriers is $N = L$, and the OFDM measurement model is stated in Section 3.4.1. We rewrite it below for convenience

$$\boldsymbol{Y_p} = \mathbf{A}\boldsymbol{h} + \boldsymbol{W_p} \tag{4.6}$$

where $\boldsymbol{Y_p}$ is the received OFDM symbols corresponding to the pilot tones and $\mathbf{A} = \mathbf{C_p}\mathbf{D_p}$. Here matrix $\mathbf{D_p}$ is the partial DFT matrix and $\mathbf{C_p}$ is a diagonal matrix with pilot symbols $\boldsymbol{X_p}$ on the diagonal elements, that is, $\text{diag}(\mathbf{C_p}) = \boldsymbol{X_p}$. This model is correct for the static channel case, however, for the time-varying channels, before the fast Fourier transform (FFT) receiving process, the received samples in the time domain is calculated by Equation 4.1, expressed as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \text{diag}\left(\underbrace{\begin{bmatrix} x_N & x_{N-1} & \dots & x_1 \\ x_1 & x_N & \dots & x_2 \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-1} & x_{n-2} & \dots & x_n \end{bmatrix}}_{\mathbf{D}^h\mathbf{C}\mathbf{D}} \cdot \underbrace{\begin{bmatrix} h_1[N] & h_1[N+1] & \dots & h_1[2N] \\ h_2[N-1] & h_2[N] & \dots & h_2[2N-1] \\ \vdots & \vdots & \ddots & \vdots \\ h_N[1] & h_N[2] & \dots & h_N[N] \end{bmatrix}}_{\mathbf{h}_{\text{full}}} \right) + \boldsymbol{w} \tag{4.7}$$

where $x_1, x_2, \dots, x_N$ is the inverse DFT (IDFT) vector of pilot symbols $\boldsymbol{X}$. After implementing DFT to Equation 4.7, we obtain the received OFDM symbols in the time-varying channel scenario as

$$\boldsymbol{Y} = \mathbf{D} \cdot \text{diag}\left(\mathbf{D}^h\mathbf{C}\mathbf{D}\mathbf{h}_{\text{full}} + \boldsymbol{w}\right) \tag{4.8}$$

Here, $\mathbf{C}$ is the diagonal matrix with all the pilot symbols on its diagonal term, the matrix $\mathbf{D}$ and $\mathbf{D}^h$ stand for the full DFT matrix and full IDFT matrix respectively, and because of the orthogonality of DFT matrix, the IDFT matrix is just the Hermitian transpose of the DFT matrix. Comparing Equation 4.8 with Equation 4.6,

we can see that the OFDM measurement model for time-varying channels can not be simply expressed by a linear equation and the linearity of Equation 4.8 is destructed by the time-varying behavior of the channel taps.

We implemented the tracking algorithms in each OFDM symbol duration and made an assumption that the channel taps stay constant during each OFDM symbol time, that is, the elements on each row of $\mathbf{h}_{\text{full}}$ are approximately the same. For all the tracking and estimation techniques, only one estimate of the channel taps will be obtained in each OFDM symbol duration, and this is main difference from the window-based tracking methods in the previous section. In this way, we can transfer the non-linear Equation 4.8 into the linear model 4.6 in each OFDM symbol duration. This is only valid for mildly varying channel, and to be specific, if we denote the coherence time of the channel as T_{coh} and the OFDM sampling duration is T_s, then $NT_s \ll T_{\text{coh}}$ need to be satisfied. It means that the OFDM symbol duration need to be much smaller than channel coherence time. As stated in Section 2.3, in some rapidly varying channels, we can not adjust the number of OFDM subcarriers since in those case N will be a negative number if $NT_s \ll T_{\text{coh}}$ is satisfied.

With the assumption mentioned above, we use the first two OFDM symbols to obtain the a priori information of the channels. We follow the constructions in [11]. Those two header symbols are composed of full pilot symbols with unit value. The estimated channel taps in the header symbols are used to acquire a first-order autoregressive (AR-1) model to capture the dynamic evolution of the channel taps. Note that only significant channel taps will be considered for building the AR-1 model. The following OFDM frames are payload frame composed of both data symbols and pilot symbols, and the Kalman filter will track the channel with the knowledge of the AR-1 model coefficients. The sparse estimation algorithms will also be implemented by solving the linear Equation 4.6. The percentage of pilot symbols in the payload frames can influence the performance of the sparse estimation algorithm significantly, however, the performance of the Kalman filter is mainly influenced by the accuracy of the AR modeling. An example of OFDM packet is shown in Figure 4.4.

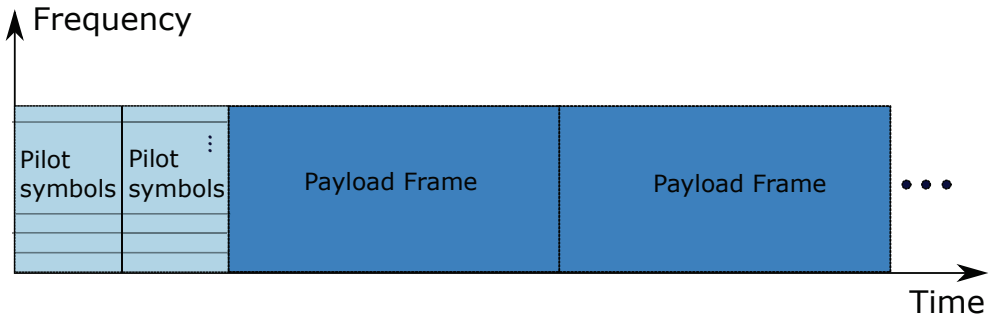


Figure 4.4: An example of OFDM packet. The first two OFDM symbols contains 100% pilot symbols and do not transmit any message. The following OFDM frames are payload frames composed of both pilot symbols and data symbols.

4.3.2 Autoregressive Modeling of Discrete Time-Varying Channel Taps

From paper [11], it has been found that the first order autoregressive model is sufficient to track the dynamic evolution of the discrete channel taps, and the number of header symbols needed is one more than the AR model order. Assume that there s significant taps among all the L taps ($s \ll L$) with index $\boldsymbol{l} = [l_1, l_2, \ldots, l_s]$, then the AR-1 model is expressed as

$$\hat{h}_{l_i}[n] = f_{l_i}\hat{h}_{l_i}[n-1] + w_{l_i}[n], \quad i = 1, 2, \ldots, s \tag{4.9}$$

where the superscript n denotes the sample index representing the time evolution, f_{l_i} denotes the AR coefficient for l_ith channel tap, and $w_{l_i}[n]$ is the uncorrelated white Gaussian noise with zero mean. With the estimation of channel taps $\hat{h}_{l_i}$ in the first two header symbols, the task now is to find f_{l_i} and the noise variance $\sigma^2_{w_{l_i}}$ for each channel tap. Finding the AR coefficients and model noise variance can be completed by constructing Yule-Walker equation first [47, 48]. Therefore, first multiplying both side of Equation 4.9 with the conjugate of $\hat{h}_{l_i}[n-1]$ and then taking the expectation operator, we obtain

$$\mathbb{E}\Big\{\hat{h}_{l_i}[n]\hat{h}^*_{l_i}[n-1]\Big\} = f_{l_i}\mathbb{E}\Big\{\hat{h}_{l_i}[n-1]\hat{h}^*_{l_i}[n-1]\Big\} + \mathbb{E}\Big\{w_{l_i}[n]\hat{h}^*_{l_i}[n-1]\Big\} \tag{4.10}$$

where the left side of Equation 4.10 is the autocorrelation of h_{l_i} with lag 1, and the right side just is the autocorrelation with zero lag since the second term is zero due to the uncorrelated noise. Therefore, we rewrite Equation 4.10 as

$$R_{l_i}[1] = f_{l_i}R_{l_i}[0] \tag{4.11}$$

where R_{l_i} stands for the autocorrelation function. Then the AR coefficient for l_ith channel tap can be solved as

$$f_{l_i} = \frac{R_{l_i}[1]}{R_{l_i}[0]} \tag{4.12}$$

The modeling noise variance $\sigma^2_{w_{l_i}}$ can be computed similarly, and the expression is

$$\begin{aligned}
\sigma^2_{w_{l_i}} &= \mathbb{E}\Big\{ w_{l_i}[n] w^*_{l_i}[n] \Big\} \\
&= \mathbb{E}\Big\{ \Big(\hat{h}_{l_i}[n] - f_{l_i} \hat{h}_{l_i}[n-1] \Big) \Big(\hat{h}^*_{l_i}[n] - f^*_{l_i} \hat{h}^*_{l_i}[n-1] \Big) \Big\} \\
&= \mathbb{E}\Big\{ \hat{h}_{l_i}[n] \hat{h}^*_{l_i}[n] \Big\} - f^*_{l_i} \mathbb{E}\Big\{ \hat{h}_{l_i}[n] \hat{h}^*_{l_i}[n-1] \Big\} - f_{l_i} \mathbb{E}\Big\{ \hat{h}_{l_i}[n-1] \hat{h}^*_{l_i}[n] \Big\} \\
&\quad + f_{l_i} f^*_{l_i} \mathbb{E}\Big\{ \hat{h}_{l_i}[n-1] \hat{h}^*_{l_i}[n-1] \Big\} \\
&= R_{l_i}[0] - f^*_{l_i} R_{l_i}[1] - f_{l_i} R^*_{l_i}[1] + f_{l_i} f^*_{l_i} R_{l_i}[0] \\
&= (1 - |f_{l_i}|^2) R_{l_i}[0]
\end{aligned} \tag{4.13}$$

where the last two step we substitute $R_{l_i}[1]$ with $f_{l_i} R_{l_i}[0]$ and apply the property that $R_{l_i}[0]$ is a real number.

To acquire the accurate value of $R_{l_i}[0]$ and $R_{l_i}[1]$, large ensemble estimation is required. However, we use only two header OFDM symbols to obtain two realization of channel taps at two consecutive OFDM symbol duration. To that end, $R_{l_i}[0]$ and $R_{l_i}[1]$ are approximated as

$$R_{l_i}[0] \approx \frac{\hat{h}_{l_i}[n] \hat{h}^*_{l_i}[n] + \hat{h}_{l_i}[n-1] \hat{h}^*_{l_i}[n-1]}{2} \tag{4.14}$$

$$R_{l_i}[1] \approx \hat{h}_{l_i}[n] \hat{h}^*_{l_i}[n-1] \tag{4.15}$$

In summary, we have obtained the AR-1 state model for formalize the dynamic evolution of the discrete channel taps. Rewriting Equation 4.9 into matrix form, we have

$$\hat{\boldsymbol{h}}_{\boldsymbol{l}}[n] = \mathbf{F} \hat{\boldsymbol{h}}_{\boldsymbol{l}}[n-1] + \boldsymbol{w}[n] \tag{4.16}$$

where $\hat{\boldsymbol{h}}[n] = \Big(\hat{h}_{l_1}[n], \hat{h}_{l_2}[n], \ldots, \hat{h}_{l_s}[n] \Big)^T$, $\hat{\boldsymbol{w}}[n] = (\hat{w}_{l_1}[n], \hat{w}_{l_2}[n], \ldots, \hat{w}_{l_s}[n])^T$, and $\mathbf{F}$ is a $s \times s$ diagonal matrix with the diagonal elements $\mathrm{diag}(\mathbf{F}) = (f_{l_1}, f_{l_2}, \ldots, f_{l_s})$.

4.3.3 Sparse-Aided Kalman Filter

The sparse-aided Kalman filter means that it only tracks the dominant channel taps whose positions can be acquired in the first two header OFDM symbols. Hence, the

OFDM measurment model becomes

$$\boldsymbol{Y_p}[n] \approx \mathbf{A_l}[n]\hat{\boldsymbol{h}}_{\boldsymbol{l}}[n] + \boldsymbol{v_p}[n] \tag{4.17}$$

where n is OFDM symbol index, and the sensing matrix $\mathbf{A_l}$ is the submatrix of $\mathbf{A}$ in the Equation 4.6 with columns $\boldsymbol{l} = [l_1, l_2, \ldots, l_s]$ included. It is of size $N_p \times s$, where N_p is number of pilot subcarriers and s is the number of dominant channel taps. Similarly, $\boldsymbol{h_l}$ is the averaged channel tap vector in each OFDM symbol duration with only dominant channel taps included and it is of size $s \times 1$. At last $\boldsymbol{Y_p}$ denotes the received OFDM symbol corresponding to the pilot tones and $\boldsymbol{v_p}$ denotes the measurement noise with zero-mean Gaussian distribution. This approximation is due to the ignorance of the non-significant channel taps.

Now we can implement the standard Kalman filter [49] running on the state-space formalism 4.16 and 4.17. There are mainly three steps in the Kalman filter recursion, prediction, Kalman gain calculation and correction, respectively. The recursion formulas are summarized in Algorithm 5.

In the prediction step, the AR-1 state model is applied to predict the channel taps in the next iteration using the estimated channel taps in the current iteration, then the prediction covariance matrix $\mathbf{C}[n|n-1]$ and Kalman gain $\mathbf{K}[n]$ are computed. Similar to the gain vector in the RLS algorithm, the larger Kalman gain indicates that more correction is needed.

4.3.4 Numerical Results

For measuring the tracking performance of sparse-aided Kalman filter, we first simulated the MSE performance in OFDM system with 12% pilot density in the payload frame. For benchmark, we also implemented OMP, LS, and RLS for comparison. Then we tried different channel variation level captured by the shape parameter $\alpha = [0.1, 1, 5]$ to observe its influence on the tracking performance.

Figure 4.5 illustrates the tracking process of all the algorithms intuitively and 4.6 presents the MSE performance of dominant channel taps. In Figure 4.5, the bold green curve represents the original time-varying channel tap and the time interval between the successive points is the sampling time. All other curves are plotted according to the OFDM symbol index, and each dot represents the estimated channel taps within

Algorithm 5 Kalman Filter

Input:

A $s \times s$ matrix $\mathbf{F}$ as diagonal coefficient matrix

A $N_p \times s$ matrix $\mathbf{A_l}[n]$ as current measurement matrix or sensing matrix

A $N_p \times 1$ vector $\boldsymbol{Y_p}[n]$ as current measurements

A $s \times s$ covariance matrix for process noise $\mathbf{W} = \mathbb{E}\left[\boldsymbol{w}[n]\boldsymbol{w}^h[n]\right]$

A $N_p \times N_p$ covariance matrix for measurement noise $\mathbf{V} = \mathbb{E}\left[\boldsymbol{v_p}[n]\boldsymbol{v_p}^h[n]\right]$

Output:

An estimate $s \times 1$ vector $\hat{\boldsymbol{h}}_{\boldsymbol{l}}[n|n]$ in each OFDM symbol duration

1: Initialization: $n = 0, \hat{\boldsymbol{h}}_{\boldsymbol{l}}[-1|-1] = \mathbf{0}_{s\times 1}, \mathbf{C}[-1|-1] = \mathbf{0}_{s\times s}$

2: **while** $n \leq$ Total number of OFDM symbols **do**

3: $\hat{\boldsymbol{h}}_{\boldsymbol{l}}[n|n-1] = \mathbf{F}\hat{\boldsymbol{h}}_{\boldsymbol{l}}[n-1|n-1]$ (Prediction)

4: $\mathbf{C}[n|n-1] = \mathbf{F}\mathbf{C}[n-1|n-1]\mathbf{F}^h + \mathbf{W}$ (Minimum prediction covariance matrix)

5: $\mathbf{K}[n] = \mathbf{C}[n|n-1]\mathbf{A}_{\boldsymbol{l}}^h[n]\left(\mathbf{V} + \mathbf{A_l}[n]\mathbf{C}[n|n-1]\mathbf{A}_{\boldsymbol{l}}^h[n]\right)^{-1}$ (Kalman gain matrix)

6: $\hat{\boldsymbol{h}}_{\boldsymbol{l}}[n|n] = \hat{\boldsymbol{h}}_{\boldsymbol{l}}[n|n-1] + \mathbf{K}[n]\left(\boldsymbol{Y_p}[n] - \mathbf{A_l}[n]\hat{\boldsymbol{h}}_{\boldsymbol{l}}[n|n-1]\right)$ (Correction)

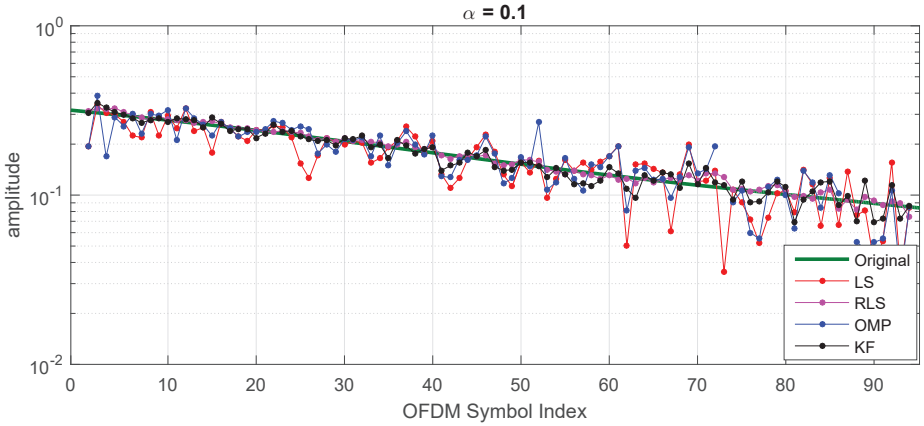
7: $\mathbf{C}[n|n] = \left(\mathbf{I} - \mathbf{K}[n]\mathbf{A_l}[n]\right)\mathbf{C}[n|n-1]$ (Minimum covariance matrix)

8: $n = n + 1$

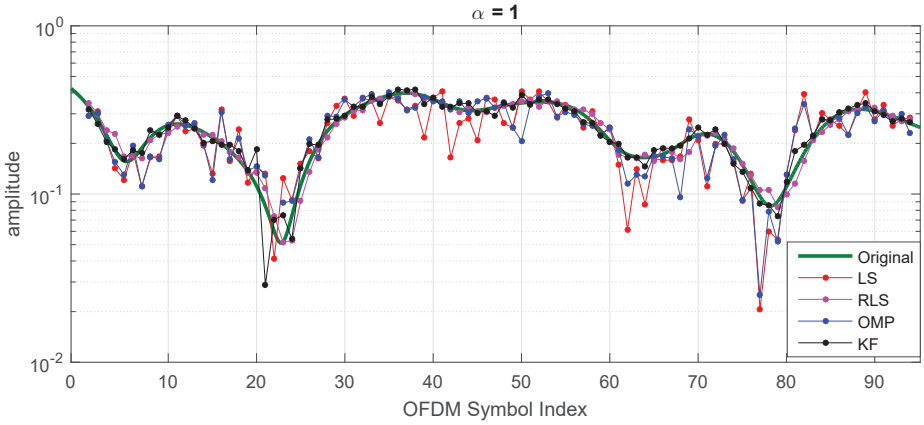
9: **end while**

one OFDM. We plot the tracking performance started from the third OFDM symbol since the first two header symbols are used for acquiring a priori information of the channel. Note that the channel taps between the dots are assumed to be constant. The number of discrete channel taps is $L = 260$ and same as the number of OFDM subcarriers $N = L$, therefore, there are 260 channel tap samples between the dots. We can see in Figure 4.6 that when α becomes larger and channel varies faster, only Kalman filter is able to keep the approximately MSE level and has the least influence of the time variation of the channel. RLS has a competitive performance compared with Kalman filter when α is small, however, its performance can be degraded as α being larger.

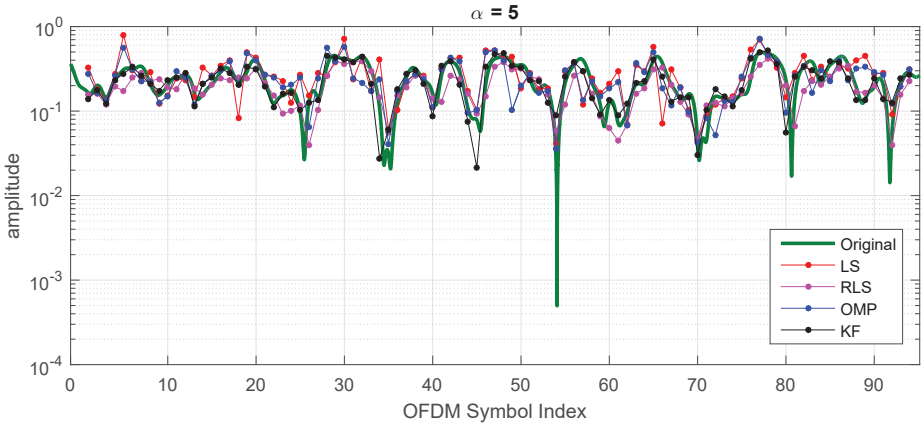
We also tested the bit error rate (BER) performance of all algorithms with 4-QAM-modulated subcarriers transmitted with bandwidth $B = 20$ kHz and carrier frequency $f = 30$ kHz. The pilot density in the payload frame we applied are [0.12, 0.16, 0.2, 0.39] respectively. The results are represented in Figure 4.7. We can see that



(a)

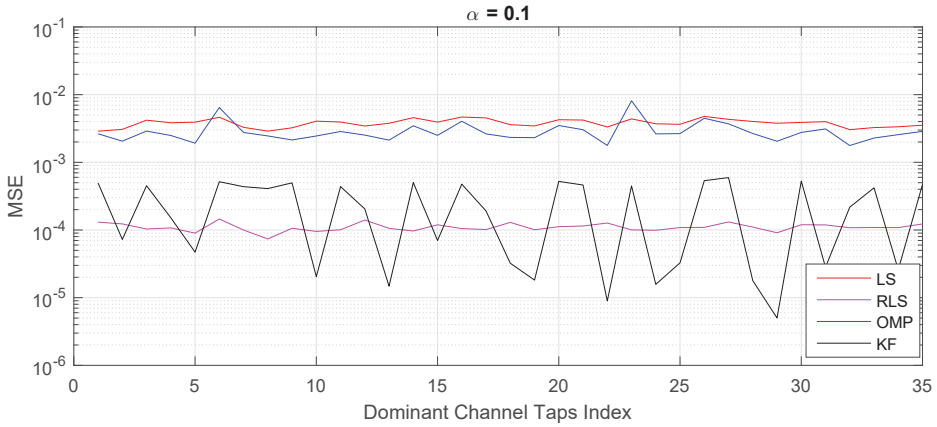


(b)

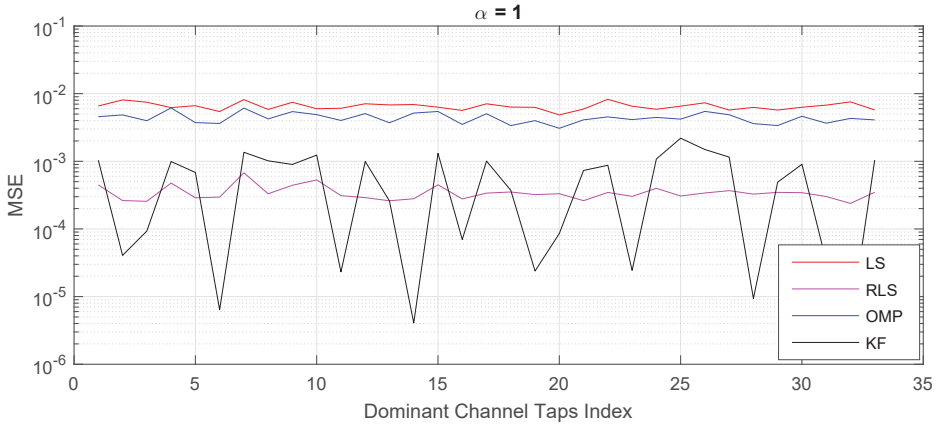


(c)

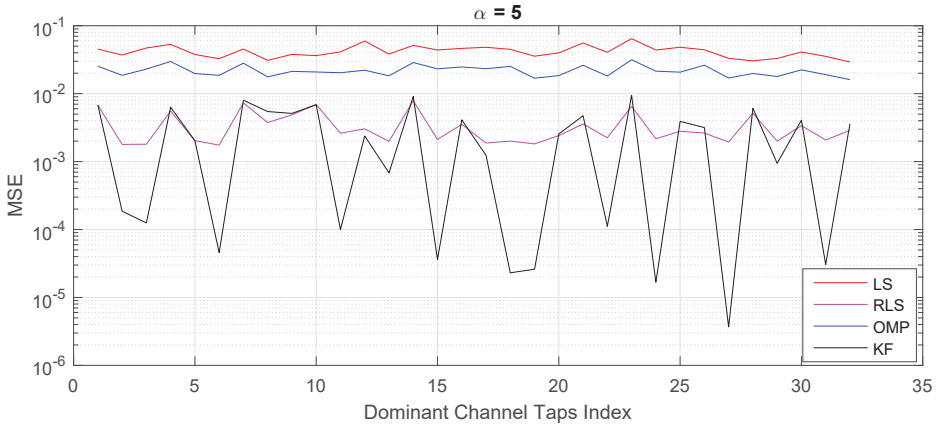
Figure 4.5: Tracking performance of sparse-aided Kalman Filter compared with RLS, OMP and LS for $\alpha = [0.1, 1, 5]$. For (a), (b) and (c), an instantaneous run for all the algorithm is implemented with SNR = 20dB and 12% pilot density in the payload frame. "KF" is short of 'Kalman Filter'".



(a)



(b)



(c)

Figure 4.6: MSE for each dominant channel tap averaged over 96 OFDM symbols. The performance of sparse-aided Kalman Filter are compared with RLS, OMP and LS for $\alpha \in [0.1, 1, 5]$, SNR = 20 dB and 12% pilot density in the payload frame.

for each value of α, the level of pilot density has very small impact on the performance of sparse-aided Kalman filter, however, it influences OMP and LS significantly. Note that the performances of all the algorithms stay almost unchanged when pilot density in the payload frame is larger or equal than 0.2, and their performances are close. However, when pilot density is smaller than 0.2, the non-tracking methods, OMP and LS suffers. We can see that the tracking algorithms, Kalman filter and RLS, require less pilot density and therefore increase the data rate. Furthermore, when $\alpha = 1, 5$ corresponding to a fast varying channel, Kalman filter is the best option for lower bit error rate, and RLS can only be competitive when $\alpha = 0.1$ corresponding to a slowly varying channel scenario.

However, observing the figures corresponding to $\alpha = 5$, we can see that Kalman filter also cannot be able to reach a lower BER for reliable communication. This is due to the invalid assumption that the channels can be approximately constant within OFDM symbol duration when $\alpha = 5$. In this case, according to the Equation 2.23 and simulations in Section 2.3, large bandwidth is required to adjusting channel coherence time within OFDM symbol duration.

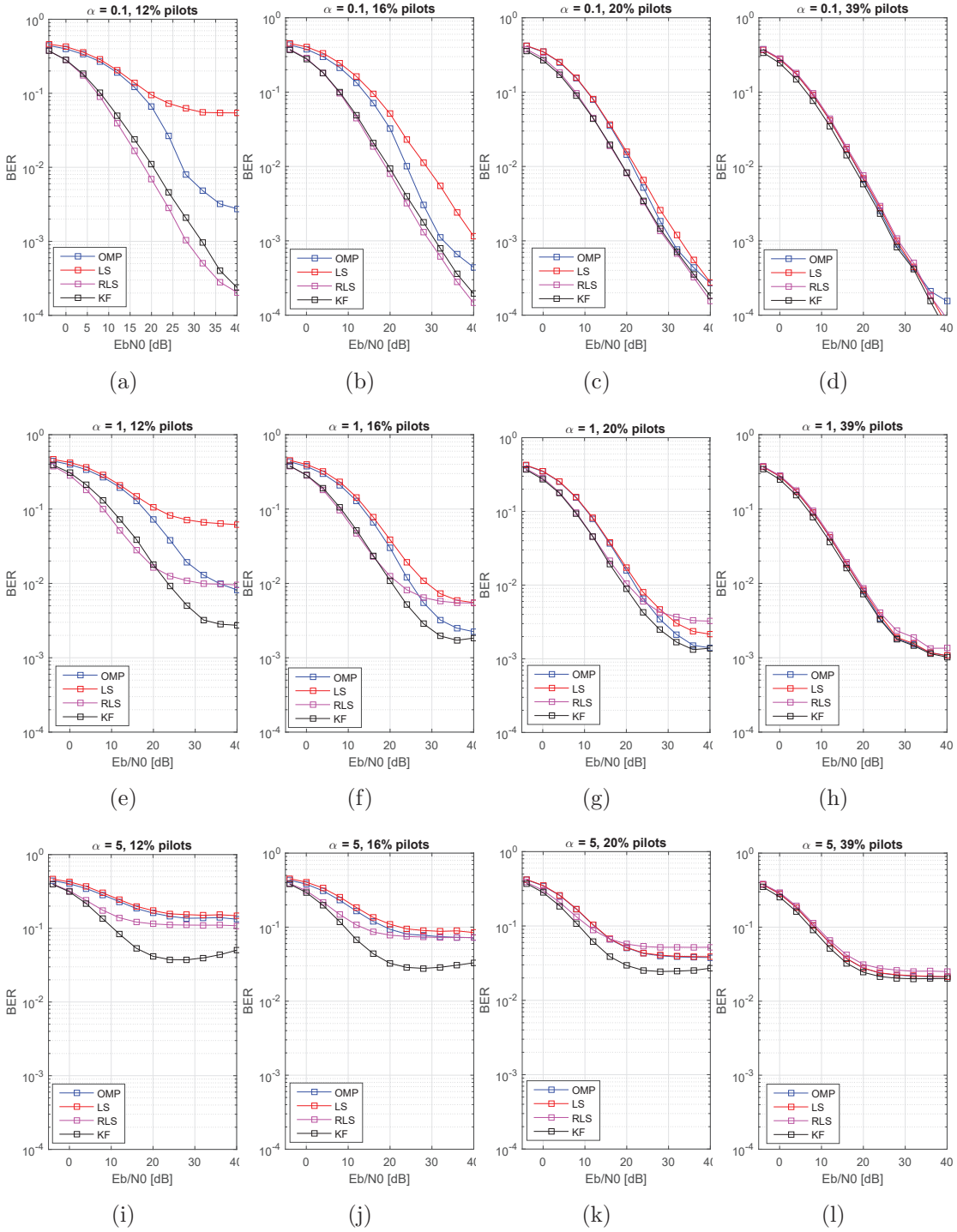


Figure 4.7: BER performance of sparse-aided Kalman filter (KF), RLS, OMP and LS with 4-QAM modulation. For each value in $\alpha \in [0.1, 1, 5]$, uncoded BER curves are simulated for pilot density (payload frame) $\in [0.12, 0.16, 0.20, 0.39]$.

Chapter 5

Analysis

5.1 Analysis of Channel Types

Based on channel studies described in Chapter 2, we find that UWA channels show different sparsity levels in different scenarios. In some cases, the channel can be very sparse and easy for sparse estimation, but there are scenarios in which the channel is dense and difficult for resolution and estimation.

For deep-water communication scenarios, the channels are usually sparse. Figures 2.12, 2.13, 2.16, demonstrate amplitude-delay profiles of such channels and we can see that there are only a few channel taps. For some values of receiver depth, the receiver could fall into a dead zone where no wave propagates according to Bellhop propagation curves and then no channel taps are observed. In addition, for medium and long range deep-water channels the channel taps are usually spaced far apart from each other in the delay domain and the super-resolution condition is typically fulfilled. The medium and long range deep-water channels may, however, be over-spread because of the long propagation delay and the resulting long multi-path profile. Even mild Doppler spread can cause the over-spread condition and only non-coherent communication is feasible in this case. For short-range communication, we can often see only one or two channel taps as shown in Figures 2.16 and 2.12. The amplitude-delay profile is shorter and in those cases, the channels are usually under-spread and may serve as a convenient medium for OFDM signaling.

For shallow-water scenarios, the channels are very different for the cases of hard bottom and soft bottom. In hard bottom cases, a lot of bounces on the bottom and surface may occur since the wave does bot lose much energy when reflected from the bottom. This makes channel multi-path profile dense and hard to estimate, especially for long-range communication (see Figure 2.14). For long range communications the super-resolution condition is usually not fulfilled in this case due to the use of low frequencies and lack of available bandwidth. Such cannels can also be over-spread.

However, if we transmit signals over a short range and the high system bandwidth, the channel can still be resolved (see Figure 2.15). Shallow-water channels with soft bottom do not have long delay profiles since bottom bounces are quickly decaying. Such channels are often under-spread in a long range communication due to mild channel variation with low carrier frequency (see Figure 2.10).

Compared with channels discussed above, the Arctic channels may be the hardest channels for communication. These channels are usually long-distance channels and the long-range communications is not easy because of the significant transmission loss due to the wave reflections on rough ice surface and bottom. In addition, the ice surface and hard ice bottom will make the channel dense (see Figure 2.19).

Hence, we conclude that medium-range (1km-10km) deep-water channels and long-range shallow-water channels (with soft bottom) maybe the best UWA channels for sparse estimation and use of OFDM systems.

5.2 Comparison between AMP and OMP Algorithms

In Chapter 3, we discussed the implementation of AMP and OMP algorithms for static (snapshot) channel estimation. Based on the phase transition results, we can see that each algorithm has its own advantages and disadvantages. AMP has been proven to be unaffected by the distribution of non-zero channel tap entries [33], but the performance of OMP can be degraded if the sparse entries have equal amplitudes (see Figure 3.3). However, OMP is easier to be implement and it usually converges faster.

We also considered channel estimation in two different settings. One is the time-domain approach where the sensing matrix $\mathbf{A}$ is a pilot-shifting matrix, and the other is the frequency-domain approach where $\mathbf{A}$ is a partial DFT matrix. In the time-domain approach, the pilot sequence we used is the Zad-off Chu sequence and it has very low autocorrelation. However, the performances of both algorithms are degraded compared to the use of classic Gaussian sensing matrix (see Figure 3.4 and 3.5) which is typically for testing of compressed-sensing algorithms. The reason is that pilot-shifting matrix does not have low cross-correlation between its columns, thus the incoherence of the matrix [19] which is related to the number of nonzero elements we can recover is at a relatively high level.

With pilot-shifting sensing matrix, OMP outperforms AMP for sparse channel cases in which the sparsity level $\delta < 0.4$. For denser channels with $\delta > 0.4$, however, AMP has better performance. This can give us an insight in which algorithm is appropriate to apply depending on the channel scenario. For example, if we communicate via a long-range shallow-water channel with a hard bottom, from Section 2.3 we know that the channel is possibly dense; therefore we need more measurements for channel estimation which means we would fall into a large δ region. Hence AMP may be a better option. However if the channel is a deep-water channel which is usually very sparse we would choose OMP.

The time domain approach discussed above is attractive for circumstances where single-carrier communications is preferable. When the channel is significantly under-spread and then the use of an OFDM communication system is preferable, and we can apply the frequency domain approach. The estimation problem formalisms are detailed in Section 3.4.1. The main advantage of the frequency domain approach is that the sensing matrix $\mathbf{A}$ is a partial DFT matrix and the algorithms have the same phase transition properties as with the Gaussian matrix (see Figure 3.8). This is due to the orthogonality property between the columns of the DFT matrix. In this case, OMP outperforms AMP for all values of under-sampling level δ. Moreover, simulations in Section 3.4.3 show that OMP also has a better MSE performance than AMP when SNR>4 and AMP may only be a good option when SNR is very low.

In summary, for a fast-varying channel which is close to be overspread we need the time-domain approach where the matrix $\mathbf{A}$ is a pilot-shifting matrix. If this channel is highly sparse, then OMP is recommended; otherwise, AMP would be a better option. On the other hand, the frequency domain approach can be implemented when the channel is under-spread and varies slower. In this case, the matrix $\mathbf{A}$ will be a partial DFT matrix and OMP algorithm is a better option to apply, unless SNR is very low.

5.3 Comparison between Sparse Channel Tracking Methods

For dynamic channel estimation, we implemented different sparse-aided tracking methods in Chapter 4. The first one is a window-based method in the time domain and the other is a frequency domain tracking approach for OFDM systems.

Just as in analysis of time domain approach in Section 5.2, in the window-based

tracking, the sensing matrix $\mathbf{A}$ in each window is a pilot-shifting matrix. The main disadvantage of this technique is that we are not able to track denser channels ($\rho > 0.1$) with small window size, based on the results of phase transition curves. However, small window size may be necessary for a fast-varying channel. Furthermore, as we know for $\delta < 0.4$, AMP cannot even operate. We have implemented and compared RLS, OMP and LS algorithms for window-based tracking, and found that RLS algorithm outperforms OMP significantly and it is only slightly influenced by the channel variation level. For all the window sizes we tested, OMP and LS can only reach the performance of RLS when $\delta = 0.39$. Here we note that RLS is a tracker, i.e., it implicitly makes use of information estimated from previous windows.

In the frequency domain tracking approach we have implemented Kalman filter, RLS, OMP and LS algorithms and we found that Kalman filter is the most powerful for all channel variation scenarios. Comparison between the algorithms show that only RLS and Kalman filter are robust with respect to variation of the pilot density, and this means that we can increase the data rate without much penalty with respect to the BER. In our setup we make symbol-based updates for the Kalman filter but consider realistic sample-to-sample channel variation. This is why we can see that the Kalman filter performance is strongly degraded when $\alpha = 5$ in Figure 4.7. In that case the channel variation during one OFDM symbol cannot be ignored.

Hence, comparing the window-based tracking and frequency domain tracking, we can see that RLS in the window-based method is the least influenced by the channel variation. The main problem of using such a approach is that incorporation of data transmission and equalization is not straightforward. On the contrary, when Kalman filter in frequency domain tracking is used, the data detection and channel equalization are easy to implement in OFDM systems, but the channel variation can have strong impact on its performance. As a consequence, for a fast-varying channel we may use the window-based tracking and for a slow-varying channel, the Kalman filter is a good option for tracking in OFDM systems.

Chapter 6

Conclusion

In this thesis, first we have introduced a basic dynamic channel model incorporating SOS model to generate the dynamic channel taps. Bellhop simulator is applied to acquire the channel multi-path profiles as inputs to the basic dynamic channel model. The speed of the channel variation is parameterized by a single value α, which is the shape parameter of Doppler spectra of the channel. In Section 2.3, we have considered a number of channel geometries and simulated the channels with different values of α. We then drew conclusions about the sparseness of the channels in various scenarios, super-resolution, and the feasibility of OFDM communication.

Then we have adapted two sparse estimation algorithms, OMP and AMP, for static channel estimation. Different tuning methods and stopping criteria are proposed. Step size and thresholding have been derived for the AMP technique. We made a detailed comparison between the estimation algorithms in terms of the derived phase transition curves and applications to various channel types.

For sparse channel tracking, we have implemented RLS algorithm for single carrier communication system and have found that the sparse-aided tracking method have better performances over all-tap tracking. RLS algorithm was also the best option compared with OMP and LS in this case. In case of OFDM communication system, we implemented sparse-aided Kalman filter as a benchmark. The Kalman filter takes into account the statistics of the channel variation and its behavior is also influenced by the accuracy of the AR-1 model of the channel variation. In our simulations, we have found that the sparse-aided Kalman filter has the best BER performance compared with OMP, LS and RLS, however, as the value of α grows, the assumption that channels stay constant during OFDM symbol is not valid any more. We can see some penalty of BER in the numerical results section.

6.1 Future Work

Regarding channel modeling, sparse channel estimation and tracking, there are several directions that we may consider for the future work:

1. Channel Modeling: We have introduced a basic dynamic channel model and only single parameter of α is used to characterize the channel variation. However, in the real world, different paths of the channel have different variation, therefore, each path needs to be characterized by a different value of α. The variation of channel tap positions should also be counted in the model. Moreover, The Doppler shift needs to be considered in the model which will introduce the time stretching effect of the transmit signal.

2. Sparse Channel Tracking: We have implemented several algorithms for tracking the time-varying channel. However, in the time domain approach, only MSE performance is tested. To verify its feasibility, how to insert data and do the equalization need to be considered.

Bibliography

[1] A Quazi and W Konrad. Underwater acoustic communications. *IEEE Communications Magazine*, 20(2):24–30, 1982.

[2] Richard K Moore. Radio communication in the sea. *IEEE spectrum*, 4(11):42–51, 1967.

[3] N Farr, A Bowen, J Ware, C Pontbriand, and M Tivey. An integrated, underwater optical/acoustic communications system. In *OCEANS 2010 IEEE-Sydney*, pages 1–6. IEEE, 2010.

[4] PA van Walree, T Jenserud, and R Otnes. Stretched-exponential doppler spectra in underwater acoustic communication channels. *The Journal of the Acoustical Society of America*, 128(5):EL329–EL334, 2010.

[5] Weichang Li and James C Preisig. Estimation of rapidly time-varying sparse channels. *IEEE Journal of Oceanic Engineering*, 32(4):927–939, 2007.

[6] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

[7] Yagyensh Chandra Pati, Ramin Rezaiifar, and PS Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44. IEEE, 1993.

[8] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on information theory*, 53(12):4655–4666, 2007.

[9] David L Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009.

[10] Arian Maleki, Laura Anitori, Zai Yang, and Richard G Baraniuk. Asymptotic analysis of complex lasso via complex approximate message passing (camp). *IEEE Transactions on Information Theory*, 59(7):4290–4308, 2013.

[11] Ulas Güntürkün, Christian Schlegel, and Dmitri Truhachev. Compression–aided kalman filter for recursive bayesian estimation of sparse wideband channels in ofdm systems.

[12] Ronald A Iltis. A sparse kalman filter with application to acoustic communications channel estimation. In *OCEANS 2006*, pages 1–5. IEEE, 2006.

[13] Avishy Carmi, Pini Gurfil, and Dimitri Kanevsky. Methods for sparse signal recovery using kalman filtering with embedded pseudo-measurement norms and quasi-norms. *IEEE Transactions on Signal Processing*, 58(4):2405–2409, 2010.

[14] Evripidis Karseras, Kin Leung, and Wei Dai. Tracking dynamic sparse signals using hierarchical bayesian kalman filters. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6546–6550. IEEE, 2013.

[15] Namrata Vaswani. Kalman filtered compressed sensing. In *2008 15th IEEE International Conference on Image Processing*, pages 893–896. IEEE, 2008.

[16] Christian Schlegel et al. *UMDCC regular report*. Ultra Maritime Digital Communication Center, 2015.

[17] Parastoo Qarabaqi and Milica Stojanovic. Statistical characterization and computationally efficient modeling of a class of underwater acoustic communication channels. *IEEE Journal of Oceanic Engineering*, 38(4):701–717, 2013.

[18] Michael B Porter. The bellhop manual and users guide: Preliminary draft. *Heat, Light, and Sound Research, Inc., La Jolla, CA, USA, Tech. Rep*, 2011.

[19] U. Güntürkün, D. Trukhachev, C. Schlegel, and D. Yin. The impact of filtering on the resolution of sparsity in channel estimation for ofdm systems. *MTS/IEEE Oceans Conference*, 2016, accepted.

[20] Ian F Akyildiz, Dario Pompili, and Tommaso Melodia. Underwater acoustic sensor networks: research challenges. *Ad hoc networks*, 3(3):257–279, 2005.

[21] John G Proakis. *Intersymbol interference in digital communication systems*. Wiley Online Library, 2003.

[22] William C Jakes and Donald C Cox. *Microwave mobile communications*. Wiley-IEEE Press, 1994.

[23] Henry Kutschale. Arctic hydroacoustics. *Arctic*, 22(3):246–264, 1969.

[24] Alexander N Gavrilov and Peter N Mikhalevsky. Low-frequency acoustic propagation loss in the arctic ocean: Results of the arctic climate observations using underwater sound experiment. *The Journal of the Acoustical Society of America*, 119(6):3694–3706, 2006.

[25] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

[26] Christian R Berger, Zhaohui Wang, Jianzhong Huang, and Shengli Zhou. Application of compressive sensing to sparse channel estimation. *IEEE Communications Magazine*, 48(11):164–174, 2010.

[27] David L Donoho, Arian Maleki, and Andrea Montanari. How to design message passing algorithms for compressed sensing. *preprint*, 2011.

[28] Arne Beurling. Sur les intégrales de fourier absolument convergentes et leur applicationa une transformation fonctionnelle. In *Ninth Scandinavian Mathematical Congress*, pages 345–366, 1938.

[29] Benjamin Franklin Logan. *Properties of high-pass signals*. 1965.

[30] David L Donoho and Benjamin F Logan. Signal recovery and the large sieve. *SIAM Journal on Applied Mathematics*, 52(2):577–591, 1992.

[31] David L Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[32] David L Donoho. For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. *Communications on pure and applied mathematics*, 59(6):797–829, 2006.

[33] David L Donoho, Arian Maleki, and Andrea Montanari. Supporting information to: Message-passing algorithms for compressed sensing. *Proc. Nat. Acad. Sci. USA*, 2009.

[34] Arian Maleki and David L Donoho. Optimally tuned iterative reconstruction algorithms for compressed sensing. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):330–341, 2010.

[35] Mohsen Bayati and Andrea Montanari. The dynamics of message passing on dense graphs, with applications to compressed sensing. *IEEE Transactions on Information Theory*, 57(2):764–785, 2011.

[36] David L Donoho, Arian Maleki, and Andrea Montanari. The noise-sensitivity phase transition in compressed sensing. *IEEE Transactions on Information Theory*, 57(10):6920–6941, 2011.

[37] David Chu. Polyphase codes with good periodic correlation properties (corresp.). *IEEE Transactions on information theory*, 18(4):531–532, 1972.

[38] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 52(2):489–509, 2006.

[39] Ulas Güntürkün. *A Study of Sparse Channel Estimation in OFDM Systems with the Tx-Rx Filters Incorporated*. Ultra Maritime Digital Communication Center, 2016.

[40] Steven M Kay. Fundamentals of statistical signal processing, volume i: estimation theory. 1993.

[41] Emmanuel J Candès and Carlos Fernandez-Granda. Towards a mathematical theory of super-resolution. *Communications on Pure and Applied Mathematics*, 67(6):906–956, 2014.

[42] Baosheng Li, Shengli Zhou, Milica Stojanovic, Lee Freitag, and Peter Willett. Multicarrier communication over underwater acoustic channels with nonuniform doppler shifts. *IEEE Journal of Oceanic Engineering*, 33(2):198–209, 2008.

[43] Christian R Berger, Shengli Zhou, James C Preisig, and Peter Willett. Sparse channel estimation for multicarrier underwater acoustic communication: From subspace methods to compressed sensing. *IEEE Transactions on Signal Processing*, 58(3):1708–1721, 2010.

[44] Philip Schniter. Low-complexity equalization of ofdm in doubly selective channels. *IEEE Transactions on Signal Processing*, 52(4):1002–1011, 2004.

[45] Jianzhong Huang, Shengli Zhou, Jie Huang, Christian R Berger, and Peter Willett. Progressive inter-carrier interference equalization for ofdm transmission over time-varying underwater acoustic channels. *IEEE Journal of Selected Topics in Signal Processing*, 5(8):1524–1536, 2011.

[46] S.S. Haykin. *Adaptive Filter Theory*. Prentice-Hall information and system sciences series. Prentice Hall, 2002.

[47] G Udny Yule. On a method of investigating periodicities in disturbed series, with special reference to wolfer's sunspot numbers. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 226:267–298, 1927.

[48] Gilbert Walker. On periodicity in series of related terms. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 131(818):518–532, 1931.

[49] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(1):35–45, 1960.